Ambulance Victoria

Solution Architecture Definition

Public Cloud Tenancy

Telstra Corporation Limited
ABN 33 051 775 556

Revision and Signoff Sheet

Change Record

Date	Version	Author	Change Description
12/09/2018	0.1	Justin Bahler	Initial Draft
10/10/2018	0.6	Justin Bahler	Updated after internal review
25/10/2018	0.7	Justin Bahler	Updated after AV feedback
31/10/2018	1.0	Justin Bahler	Included system context diagram. Approved at STAGG

Reviewers

Date	Version	Reviewer	Company	Position
10/10/2018	0.4	Vikas Madan	Kloud Solutions	Infrastructure Consultant
11/10/2018	0.6	Hasabnis, Mansi	Ambulance Victoria	
11/10/2018	0.6	Pham, John	Ambulance Victoria	
11/10/2018	0.6	Sharma, Rahul	Ambulance Victoria	
11/10/2018	0.6	Tan, Seng Hee	Ambulance Victoria	
11/10/2018	0.6	Vandromme, Sven	Ambulance Victoria	
11/10/2018	0.6	Trush, Elena	Ambulance Victoria	
11/10/2018	0.6	Romanis, Paul	Ambulance Victoria	
11/10/2018	0.6	Muller, Jason	Ambulance Victoria	
11/10/2018	0.6	Pietrzykowski, Dominik	Ambulance Victoria	
11/10/2018	0.6	Reynolds, Gary	Ambulance Victoria	
11/10/2018	0.6	Smith, Paul	Ambulance Victoria	
11/10/2018	0.6	Mark Simmonds	Ambulance Victoria	
11/10/2018	0.6	Falkinder, Sam	Ambulance Victoria	
11/10/2018	0.6	Sala, Sam	Ambulance Victoria	
11/10/2018	0.6	Howarth, Daniel	Ambulance Victoria	
11/10/2018	0.7	STAGG	Ambulance Victoria	

Table of Contents

1 Introduction 5
1.1 Overview 5
1.2 Document Purpose 5
1.3 Document Scope 5
1.4 Document Audience 5
1.5 Document Conventions 5
1.6 Related Documents 5
2 Solution Overview 7
2.1 High-level Solution Description 7
2.2 System Context Diagram 7
3 Environments 8
3.1 Environment Design 8
3.2 Environment Consumption 9
3.3 Environment OLAs 9
3.4 Environment Maintenance Windows 10
4 Azure Hierarchy 12
4.1 Subscription Partitioning 12
4.2 Azure Regions 14
4.3 Resource Groups 14
4.4 Resource Tagging 15
4.5 Cost Management 15
4.5.1 Reserved Instances 16
5 Networking and Connectivity 17
5.1 Wide Area Network (WAN) Design 17
5.1.1 Overview 17
5.1.2 Telstra Cloud Gateway 18
5.1.3 ExpressRoute Logical Connectivity 19
5.1.4 Internet Connectivity 22
5.2 Azure Internal Network Design 26
5.2.1 Overview 26
5.2.2 Virtual Networks 28
5.2.3 Virtual Network Peering 28
5.2.4 Virtual Network Gateway 28

5.2.5 Subnets 29
5.2.6 Network Security Groups 30
5.2.7 Virtual Network Service Endpoints 31
5.2.8 Load Balancers 31
5.2.9 DNS Services 32
5.2.10 User-Defined Routes 32
6 Azure Platform Components 33
6.1 Storage 33
6.1.1 Storage Accounts 33
6.1.2 Storage Account Access Control 34
6.1.3 Storage Account Encryption 35
6.1.4 Managed Disks 35
6.2 Monitoring and Alerts 36
6.3 Backup and Recovery 36
6.3.1 Commvault 37
6.3.2 Azure Backup 37
6.3.3 Azure Site Recovery Service 38
6.4 Azure SQL Database 42
6.4.1 Azure SQL Replication 43
6.5 Azure Automation 43
7 Security Components 45
7.1 Azure Security Center 45
7.2 Role Based Access Control (RBAC) 46
8 Identity and Access Management 49
8.1 Overview 49
8.2 Active Directory 49
8.2.1 AD Sites & Services 50
8.3 Azure Active Directory 50
8.4 AADC 51
8.5 ADFS 51
8.6 Azure Enterprise Applications 51
8.7 MFA 52
9 Application Hosting Services 53
9.1 Azure Virtual Machines 53
9.1.1 Virtual Machine Specifications 53
9.1.2 Virtual Machine Operating Systems 53

9.1.3 Azure Virtual Machine Extensions for Windows 54
9.1.4 Azure Virtual Machine Configuration for Linux 54
9.1.5 Virtual Machine Antimalware Protection 54
9.2 Azure App Service 55
9.2.1 Azure Application Service Environments 56
9.3 Azure Functions 56
9.4 API Management 57
9.5 Azure Service Fabric 57
10 Deployment Tools 59
10.1 Overview 59
10.2 Infrastructure Coding Language 59
10.3 Visual Studio Team Services 60
10.3.1 Overview 60
10.3.2 Code Repository and Source Control (Git) 60
10.3.3 Build and Release 61
10.3.4 Integration 63
Appendix A - Requirements Traceability 65
All Requirements by Status 65
All Requirements by Criticality 65
Requirements Traceability Matrix 65
Appendix B - Contents Tables 72
Table of Key Design Decisions 72
Table of Recommendations 73
Table of Figures 73

1 Introduction

1.1 Overview

The Solution Architecture Definition (SAD) is a set of high-level design decisions laying out the plan for all elements of Ambulance Victoria's Public Cloud Tenancy project to be implemented. It contains explanations of concepts, solution elements and design decisions as well as recommendations for future requirements.

1.2 Document Purpose

The purpose of this document is to give a detailed design of the solution being implemented as part of the Public Cloud Tenancy project. It includes four main system elements; Microsoft Azure tenancy architecture, Azure infrastructure design, network connectivity design and solution design for the applications being migrated to the Azure platform. The design document will be relatively high-level as most of the system configuration details will be held in the 'As-Built' document.

1.3 Document Scope

The scope of this document is limited to infrastructure and networking components built and deployed to support the Public Cloud Tenancy project, test application workloads and the implementation of additional networking components. While this document may cover or touch upon some application level elements, the intention is to provide context and in some cases justification for an infrastructure component being deployed and hence should be used as reference only and not as the source of truth. For detailed information on the deployed components, refer to the 'As-Built' documentation.

1.4 Document Audience

The intended audience for this document is Ambulance Victoria personnel involved in the Public Cloud Tenancy project and any other staff who will be working with either Azure resources or solutions integrated or deployed on Azure (including 3rd party or contracting staff). This document contains technical concepts and is targeted at those staff with a strong working technical knowledge.

1.5 Document Conventions

Key Design Decision
Throughout this document you will find these "Key Design Decision" boxes. Each of these represents a choice put to the team regarding the design of the solution. Each design decision has been taken with either direct consultation with the internal Ambulance Victoria IT team or members of the on-site Kloud Consulting team. These boxes will explain the decided route and the reason for choosing it.

Recommendation
In addition to Key Design Decisions, recommendations will be captured in "Recommendation" boxes throughout the document. Each of these represents a recommendation from the team for future use and adoption of the solution but not delivered as part of the solution.

1.6 Related Documents

Document Name	Description	Link

AV - Public Cloud Tenancy - Requirements Document v1.0	Agreed solution requirements	TBA
AV - Public Cloud Tenancy - Application Deployment Patterns v1.0	Reference architecture document containing HA and DR deployment patterns for Platinum services	TBA

2 Solution Overview

2.1 High-level Solution Description

The Public Cloud Tenancy solution involves the following:

- Two virtual datacentres that are Azure cloud based to deploy virtual machines and Web Apps in.
- Four cloud deployment patterns for Ambulance Victoria's services that are defined as Platinum.
- Azure cloud-based disaster recovery option to enable regional redundancy for on-premise Platinum workloads.
- Active Directory Domain Controllers deployed into Azure IaaS as the identity management suite to enable single sign-on for IT staff, domain joining servers and local DNS services.
- Infrastructure deployed through Visual Studio Team Services (VSTS), to provide repeatable, consistent, agile and speedy delivery of application platforms.
- A VSTS Git based code repository for infrastructure ARM templates and PowerShell scripts, so code can be worked on simultaneously and code merges can be seamlessly merged into the central repository.
- Integration into existing Ambulance Victoria services for event management, security update management, Active Directory

2.2 System Context Diagram

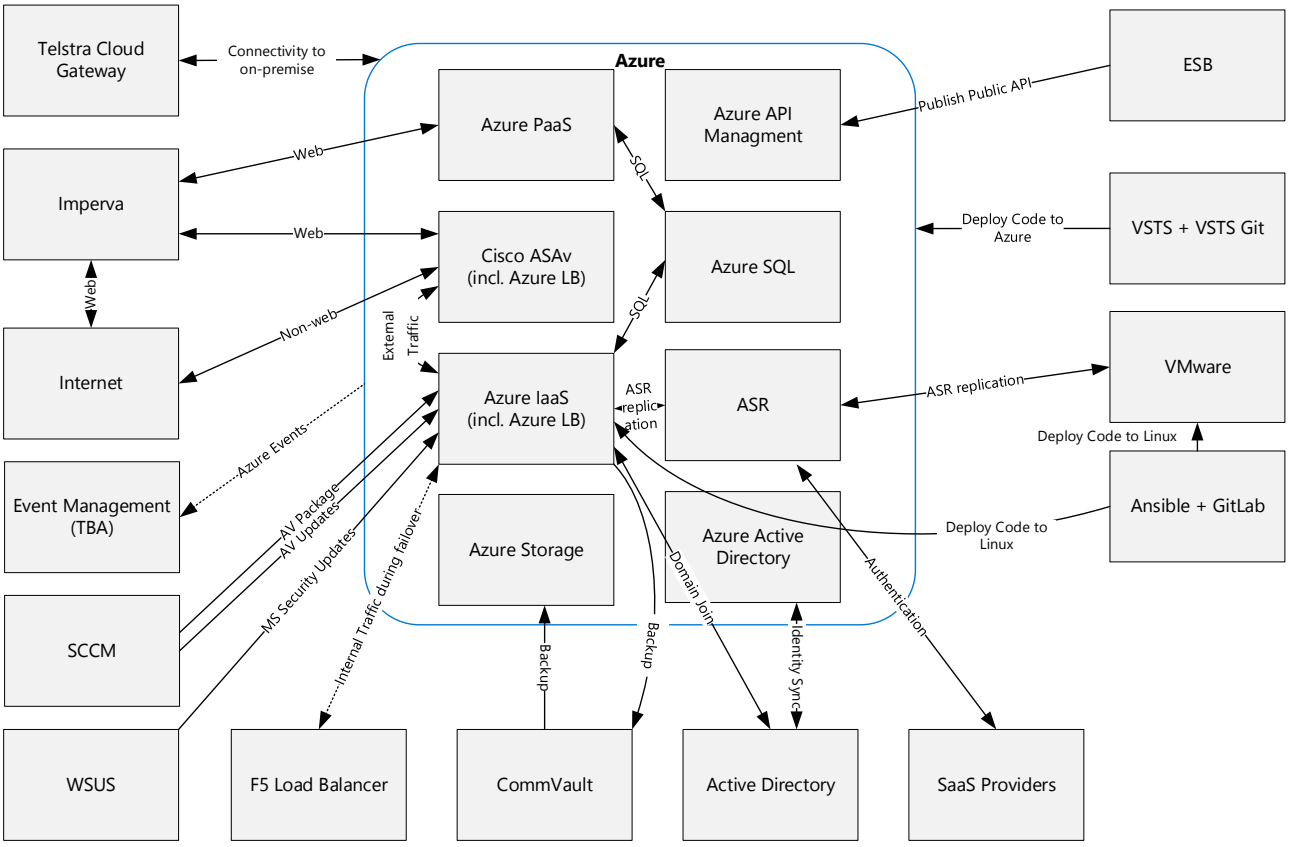

Figure 1 - System Context Diagram

3 Environments

3.1 Environment Design

The purpose of this section is to define the environments based on Ambulance Victoria's release management processes. Defining this will align the foundation architecture to how Ambulance Victoria release software and changes into the production environment.

An environment can be described as a controlled and often repeatable Configuration or set of Configurations that are perceived to act as a contained, bordered or surrounding operational context and that allow one or more Entities such as Resources or Systems to perform one or more controlled functions or activities. In an enterprise IT organisation these can often be referred to as the logical and/or physical container for development, test and production workloads.

Ambulance Victoria's current environmental design is application dependant and applied in an inconsistent way e.g. for Oracle environments there is development, test and production environments whereas for others there is only test and production. The environments are also not segregated physically by firewalls or access controls and they all use production shared services like the Active Directory and Azure Active Directory. This presents a risk of development work impacting production and/or slowing down development as production-like due diligence is needed in high-change development phase. e.g. schema upgrades needed as part of an application development.

The only logical separation is via different server names for production vs non-production workloads and deployment processes.

Key Design Decision

There is currently only a single Azure Active Directory tenancy, it is prudent to establish an additional environment to allow proof-of-concept (PoC) creation and testing of core Azure AD and Azure features in an isolated environment that would not have a possibility of impacting production.

Deployment Environments

The environment setup will be as follows:

Environment	Abbreviation	Description
Sandbox	SBX	Isolated environment for testing PoC's of new Azure capability without impacting existing environments. (e.g. to enable and test Azure AD Domain Services etc)
Development	DEV	Application development environment
Test	TST	Pre-release technical testing environment
User-acceptance test	UAT	Environment for end user testing
Production	PRD	Business critical production environment
Disaster Recovery	DR	Disaster recovery environment to allow failover of the service between regions

Environment Classification

The environments will be classified into non-production, production and sandbox. "Production" for services where users will be consuming the service, "Non-production" where only development and testing users will be utilising the services and "Sandbox" where only isolated PoC's can be developed and tested.

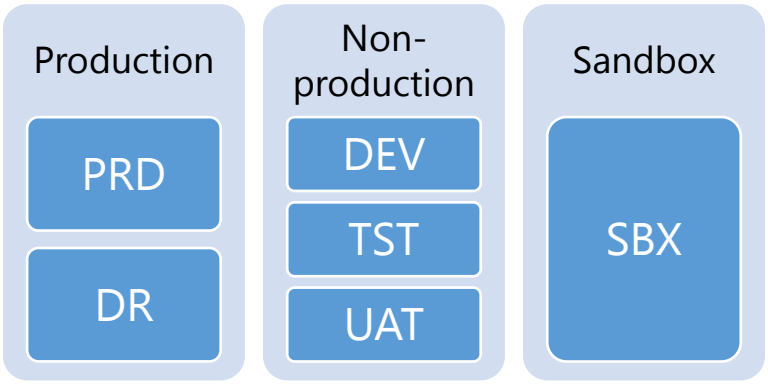

Figure 2 - Environment Classification Mapping

Key Design Decision 3-1: Environment Design

3.2 Environment Consumption

The pattern for consumption of the environments will vary slightly depending on whether Ambulance Victoria will be coding a new application and managing the Software Development Lifecycle (SDLC).

Ambulance Victoria performing or purchasing the coding

Ambulance Victoria purchasing Commercial-off-the-Shelf (COTS) applications

Proof-of-Concept

3.3 Environment OLAs

An operational-level agreement (OLA) defines the interdependent relationships in support of a service-level agreement (SLA). The agreement describes the responsibilities of each internal support group toward other support groups, including the process and timeframe for delivery of their services. This information will help with designing the components required to meet the OLA requirements for each environment.

Ambulance Victoria's Service Offering support matrix[1] is extracted below. Key elements from the table are the 'RPO' for backup frequency, 'RTO' for speed of recovery, 'Availability' for underpinning component availability requirements and 'Initial Acknowledgement' for assisting in monitoring requirements:

[1] AV ICT Business Service Catalogue v1.0

Service Type	Support Coverage	Availability	Priority	Initial Acknowledgement and Updates	Restoration or Resolution Targets	RTO	RPO
Platinum	24x7 or Business Hours	99.90%	P1	30 minutes (24x7)	4 Hours (24x7)	~0 Hours	~0 Hours
			P2	1 hour (24x7)	8 Hours (24x7)		
			P3	1 day (business hours)	3 Business Days		
			P4	3 days (business hours)	5 x Business Days		
Gold	24x7 or Business Hours	99.50%	-	-	-	4-24 Hours	~0-4 Hours
			P2	1 hour (24x7)	8 Hours (24x7)		
			P3	1 day (business hours)	3 Business Days		
			P4	3 days (business hours)	5 x Business Days		
Silver	Business Hours Only	98.00%	-	-	-	1-3 Days	1-2 Days
			P2	1 hour (24x7)	8 Hours (24x7)		
			P3	1 day (business hours)	3 Business Days		
			P4	3 days (business hours)	5 x Business Days		
Bronze	Business Hours Only	N/A	-	-	-	7 Days	2 Days
			-	-	-		
			P3	1 day (business hours)	3 Business Days		
			P4	3 days (business hours)	5 x Business Days		

Table 1 - Ambulance Victoria's Service Offering support matrix

Key Design Decision
To cater for one of the primary strategic use cases, Cloud HA/DR, the application deployment patterns will assess and design the underpinning infrastructure's availability to be as fault tolerant as possible to meet the 99.90% availability and ~0 Hours RTO/RPO metrics for Platinum services.

Key Design Decision 3-2: Environment OLAs

3.4 Environment Maintenance Windows

Microsoft Windows servers require regular patching that needs to be scheduled as the patching process often requires a reboot. To facilitate this a weekly maintenance window will be established which is a known period that the system "can" be taken offline due to limited use, if required.

Key Design Decision
The production environment maintenance window could be Tuesday-Thursday at 0200 to 0500. This is outside core hours and enables enough time to complete any required maintenance. Patching can be scheduled to be completed on the systems in a staggered approach as follows:

Server	Schedule	Notes
Management Servers	Tuesday-Thursday 0200	Least critical to production service

		Immediate health check to confirm working
Web server 1	Tuesday-Thursday 0300	Will be servicing half the inbound load Immediate health check required to validate continuing with the remaining patches
Web server 2	Tuesday-Thursday 0400	
SQL server 1	Tuesday-Thursday 0300	
SQL server 2	Tuesday-Thursday 0400	

Key Design Decision 3-3: Environment Maintenance Windows

4 Azure Hierarchy

4.1.1 Subscription Partitioning

Microsoft Azure agreements have four levels of hierarchy; Enterprise Enrolments, Departments, Accounts and finally Subscriptions. Azure enrolment hierarchies define how services are structured within an Enterprise Agreement. The Enterprise Portal allows customers to divide access to Azure resources associated with an Enterprise Agreement based on flexible hierarchies customizable to an organization's unique needs. The hierarchy pattern should match an organization's management and geographic structure so that the associated billing and resource access can be accurately accounted for. The three high-level patterns are functional, business unit, and geographic, using departments as an administrative construct for account groupings. Within each department, accounts can be assigned subscriptions, which create silos for billing and several key limits in Azure (e.g., number of VMs, storage accounts, etc.)[2].

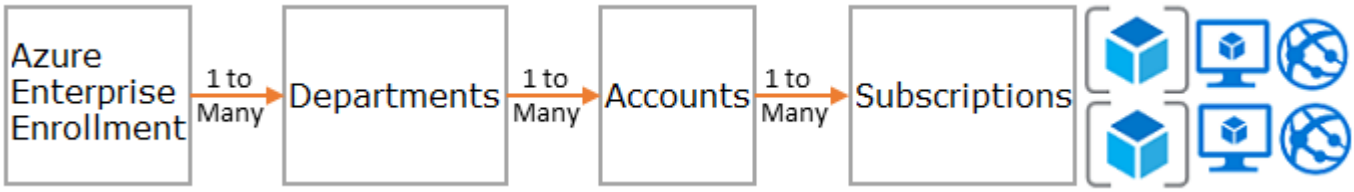

Figure 3 - EA Enrolment Infographic

Enterprise Enrolment

The root level element of governance, tied to an Azure Enterprise Agreement. May contain multiple departments, accounts and/or subscriptions.

Departments

A Department is an administrative division of organizational hierarchy, based on the selected hierarchy pattern (functional, geographical or business unit). Departments can be created within the Enterprise Agreement Portal as a mechanism for associating one or more Accounts for billing and reporting purposes. Within an EA spending limits and alerts are assigned at a Department level within the Azure Hierarchy rather than at the subscription level like PAYG or MSDN accounts.

Accounts

An Account is a logical container in which Subscriptions are created. A single Account Owner is defined for each Account who is the responsible entity for creation, cancellation and general management of all associated Subscriptions. The Account Owner is an individual or group associated with an email address, which may belong to either an Azure AD account or Microsoft account. It is also given a descriptive name within the Enterprise Portal for administrative purposes. Holds one or more Azure subscriptions.

Subscriptions

Subscriptions are logical entities which enable the organising and deployment of Microsoft Azure services. Subscriptions are also the billing and reporting container for the associated deployed services, and for this reason, it is commonly considered good practice to distribute unrelated services amongst multiple Subscriptions as opposed to combining services in a single Subscription to ensure that the service costs can be easily and concisely reported and understood.

[2] https://www.credera.com/blog/credera-site/azure-governance-part-1-understanding-the-hierarchies/

Several other benefits are realised by segregating services across multiple Subscriptions, including:

- Nominated Subscriptions can be transferred to different Account Owners.
- Nominated Subscriptions can be transferred to Accounts associated with alternative Enterprise Agreements.
- Nominated Subscriptions can be transferred to different payment mechanisms if required.
- Nominated Subscriptions and associated services can be cancelled if required.

Key Design Decision
Within Ambulance Victoria the Azure hierarchy will have one Enterprise Enrolment but will require two Departments. One for Production and Non-Production resources and a second Department to contain a Sandbox subscription. The reason for the second department is that under an EA Enrolment it is only possible to place spending limits and alerts at the Department level. There will be one Account for each Department. Finally, there will be 3 subscriptions, two under the main Ambulance Victoria IT department (one for production and shared resources and a second for non-production resources) and one under the Ambulance Victoria Testing department which will be a sandbox for infrastructure and development staff to use for testing out new solutions and components. Some thought went into deciding whether to create multiple subscriptions for each of the different environments, solutions or departments, however, Ambulance Victoria have decided that it will be simpler to keep to just the three subscriptions and use network level segregation to separate environments, Resource Groups for permissions and Azure resource tagging for billing.

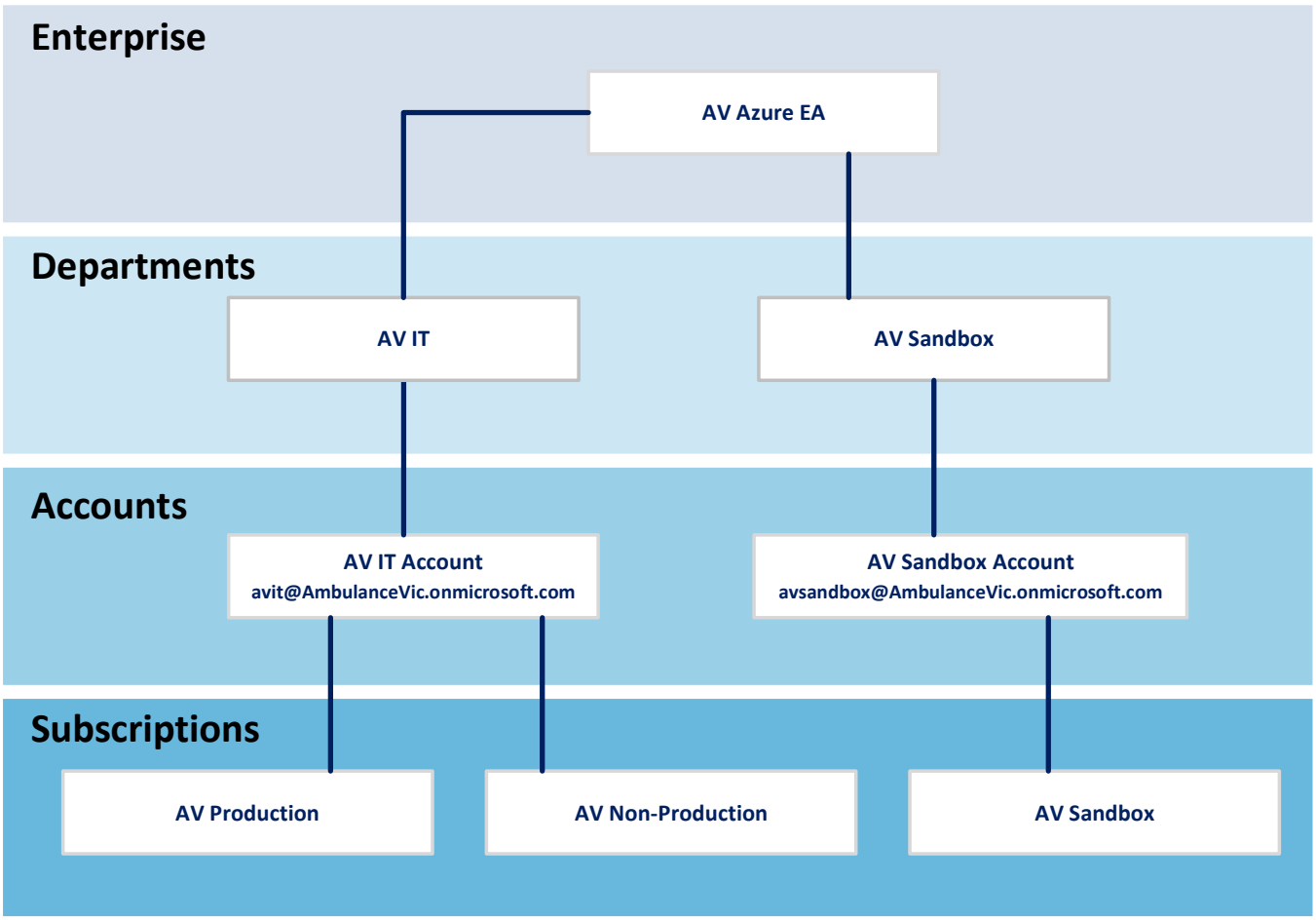

Figure 4 - Ambulance Victoria Azure EA Structure

Key Design Decision 4-1: Subscription Partitioning

4.2 Azure Regions

Ambulance Victoria's Azure infrastructure will make use of two regions; Australia Southeast (Melbourne) for all production and non-production resources and Australia East (Sydney) for any DR and Backup resources. As part of Ambulance Victoria's data sovereignty compliance all data must be stored on Australian deployed servers and components. Some data processing components such as Machine Learning workspaces or Data Factories may be deployed in other regions as they are currently unavailable in Australia.

Key Design Decision
To allow for regional redundancy the following Azure Region layout will be utilised: Australia Southeast – Production, Non-Production, Sandbox Australia East – DR Ambulance Victoria current fall under the TPAM[3] agreement, further due diligence needs to be performed to assess any impact to the agreement when having DR run out of the Sydney based Azure datacenter.

Key Design Decision 4-2: Azure Regions

4.3 Resource Groups

Microsoft defines Resource Groups as a container which holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources which need be managed as a group[4].

There are some key factors to consider when defining resource groups:

- All the resources in your group should share the same lifecycle. You deploy, update, and delete them together. If one resource, such as a database server, needs to exist on a different deployment cycle it should be in another resource group.
- Each resource can only exist in one resource group.
- You can add or remove a resource to a resource group at any time.
- You can move a resource from one resource group to another group.
- A resource group can contain resources that reside in different regions.
- A resource group can be used to scope access control for administrative actions.
- A resource can interact with resources in other resource groups. This interaction is common when the two resources are related but do not share the same lifecycle (for example, web apps connecting to a database).

Key Design Decision
For the Public Cloud Tenancy Project, Resource Groups will be used for a combination of cost control and Role Based Access Control (RBAC). Resource groups will be created to house the following as required: Solution/Project specific resources (e.g. an internal web portal) Platform Components (e.g. an Enterprise Data Management Tool) Shared Resource Components (like networking) Each resource group will have permissions applied to allow internal Ambulance Victoria teams to manage their own resources. Resource Groups should also have Azure tags applied (and tags should be automatically applied to all resources within the group) to allow for more granular cost reporting. Detailed tagging standards can be found in the 'Standards and Governance' document.

[3] http://www.procurement.vic.gov.au/State-Purchase-Contracts/Telecommunications-TPAMS2025-Services

[4] https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-overview

Resource Group security needs to be split based on application (e.g. internal web portal app) and by function (e.g. so a team could manage all the backup servers or networking resources). More detailed information can be found in the Role Based Access Control (RBAC) section of this document.

Key Design Decision 4-3: Resource Groups

4.4 Resource Tagging

The Azure platform enables Ambulance Victoria to logically organise resources by applying tags. The tags consist of key/value pairs that identify resources with properties that Ambulance Victoria can define.

When viewing resources with a specific tag it is possible to see resources from all their resource groups. The benefit of defining and applying a tag taxonomy is to provide the capability to logically organise resources in a way that is independent of the deployment relationships. A common usage of a tag taxonomy is to group resources for billing or management.

Each tag defined for an individual resource or within a resource group is automatically added to the subscription-wide taxonomy. Ambulance Victoria can also prepopulate the taxonomy with tag names and values likely to be consumed in the future.

Key Design Decision
To maximise the opportunities to report based on costs and ownership, all Azure assets will be tagged. Tagging will be applied at the Resource Group level so all resources within the group have the same tags. This approach minimises administrative overhead. The only exception is for tags relating to the power cycles of virtual machines, these tags will be manually applied to the virtual machine and not at the resource group level. Component tags will be copied from the Resource Group tags by a nightly automation task.

Key Design Decision 4-4: Resource Tagging

4.5 Cost Management

Azure Cost Management licensed by Cloudyn, a Microsoft subsidiary, allows you to track cloud usage and expenditures for your Azure resources and other cloud providers including AWS and Google. Easy-to-understand dashboard reports help with cost allocation and showbacks/chargebacks as well. Cost Management helps optimize your cloud spending by identifying underutilized resources that you can then manage and adjust.[5]

Key Design Decision
Cloudyn will be integrated with Azure to assist with cost management and provide easy to use dashboards and reports for Ambulance Victoria staff.

Key Design Decision 4-5: Cost Management

Recommendation
Cloudyn can be further utilized to centralise reporting and also perform AWS cost management. This is an additional cost to the free pricing tier which only allows for Azure reporting. If Cloudyn does not provide the dashboards and reports required, Ambulance Victoria can utilise Power BI to integrate with Azure billing APIs and create custom dashboards/reports to meet their business requirements.

Recommendation 4-1: Cost Management

[5] https://docs.microsoft.com/en-us/azure/cost-management/overview

4.5.1 Reserved Instances

Azure Reservations[6] helps you save money by pre-paying for one-year or three-years of virtual machine, SQL Database compute capacity, Azure Cosmos DB throughput, or other Azure resources. Pre-paying allows you to get a discount on the resources you use. Reservations can significantly reduce your virtual machine, SQL database compute, Azure Cosmos DB, or other resource costs up to 72% on pay-as-you-go prices. Reservations provide a billing discount and don't affect the runtime state of your resources.

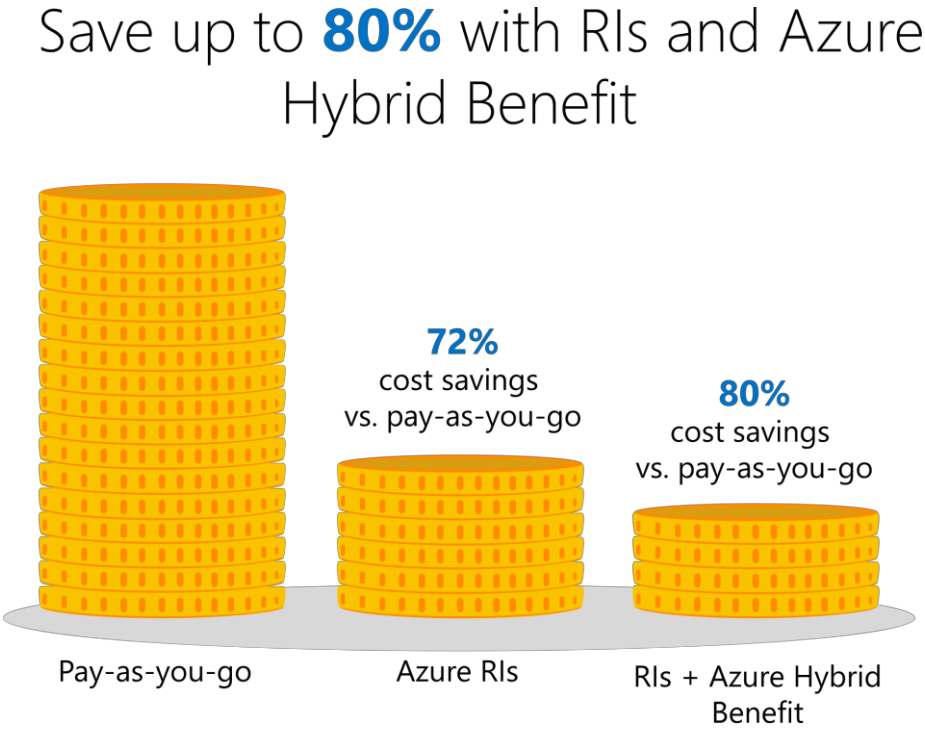

Figure 5 - Azure Reserved Instances Benefits Diagram

Key Design Decision
Reserved instances will be utilized for production workloads that are determined to remain online for more than a year e.g. Active Directory Domain Controllers

Key Design Decision 4-6 - Reserved Instances

[6] https://docs.microsoft.com/en-us/azure/billing/billing-save-compute-costs-reservations

5 Networking and Connectivity

5.1 Wide Area Network (WAN) Design

5.1.1 Overview

One of the key requirements for the solution is to treat Azure as if it were the same as an existing "on-premise" datacentre. This means that dedicated links between the Ambulance Victoria sites and Azure need to be established. These links should enable seamless integration between all sites.

Ambulance Victoria also has a requirement for redundant links to reach the new Azure environment. Requirements were gathered over several meetings, and it was determined that Ambulance Victoria would like dual virtual connections to Azure in the form of Cloud Gateways. Regional redundancy is also required, in the event the entire Melbourne region goes offline the Azure environment should still be reachable from the internal network.

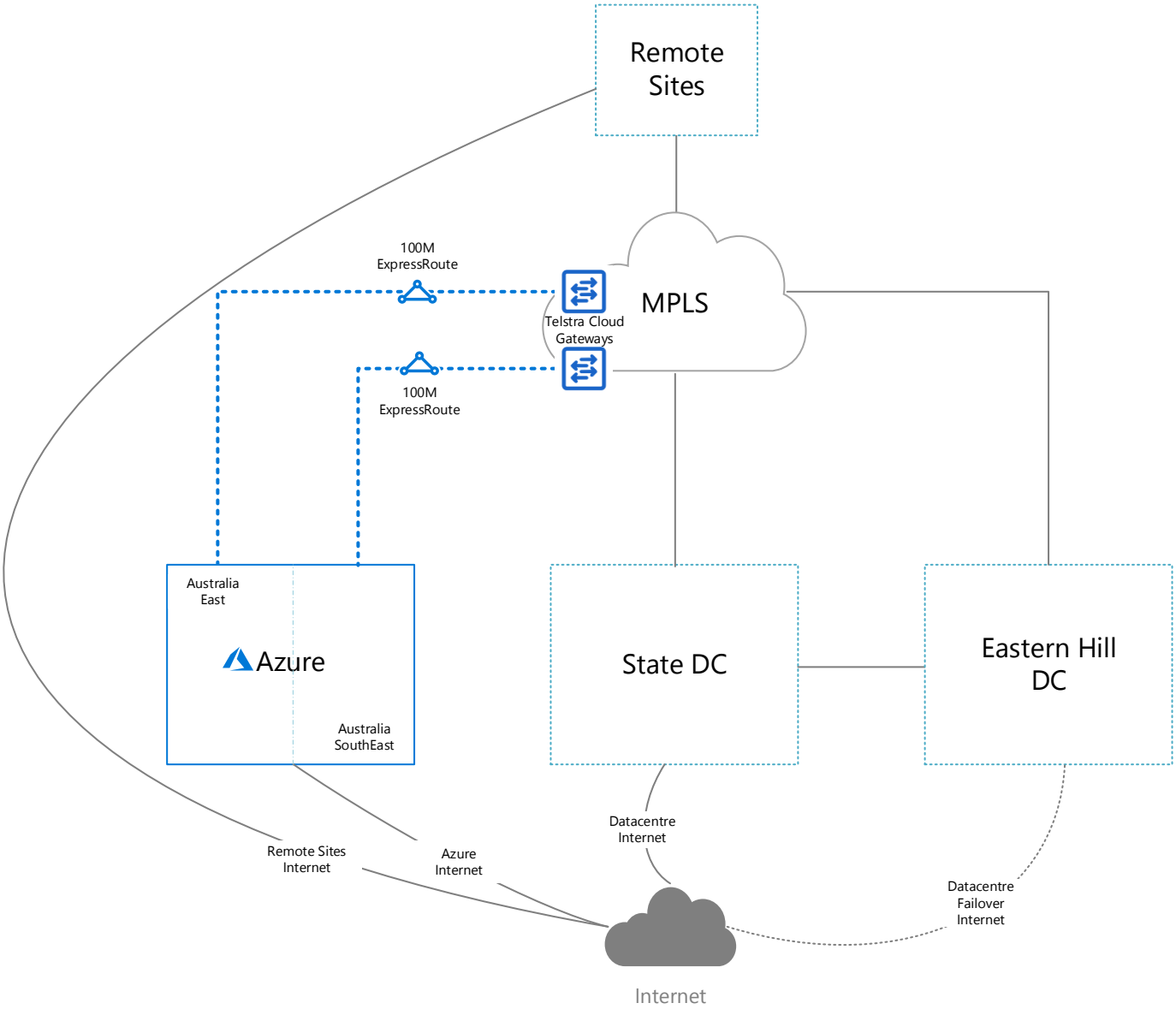

Figure 6 – Proposed Wide Area Network Overview

Key Design Decision
Two dedicated links will be configured to connect Azure and the existing network. Overlaid on those links will be two ExpressRoute links. The primary link will terminate in the Azure Australia Southeast (Melbourne) region and secondary link will terminate in the Australia East (Sydney) region.

Key Design Decision 5-1: Wide Area Network

5.1.2 Telstra Cloud Gateway

5.1.2.1 Overview

The Telstra Cloud Gateway[7] service offers a simple way to connect an existing on-premises network to one or multiple public cloud or Telstra cloud services. It can connect to the following services:

- Microsoft Azure and Office 365
- Amazon Web Services
- VMWare Cloud Air
- IBM Cloud
- Oracle Cloud

The Cloud Gateway can be attached to an existing Telstra MPLS network allowing for extension of on-premises networks to public cloud infrastructure. The gateway is capable of being connected to multiple services at once, for example, it is possible to have direct links to Azure, AWS and the Oracle Cloud simultaneously from the same Telstra Cloud Gateway service.

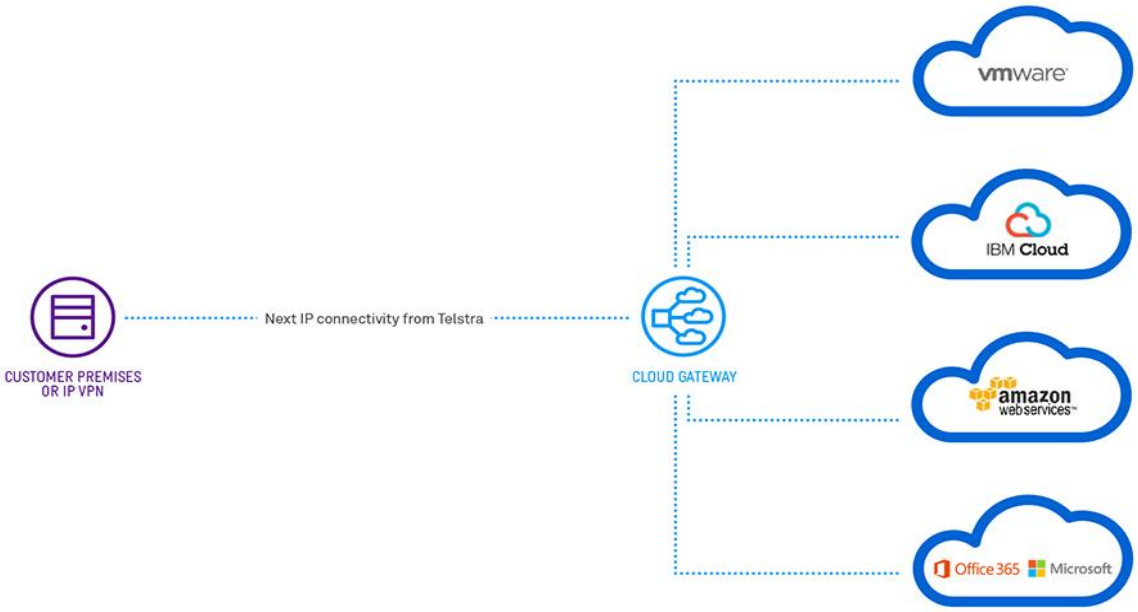

Figure 7 - Telstra Cloud Gateway connectivity options

[7] https://cloud.telstra.com/res/pdf/cloud-gateway-technical-guide.pdf

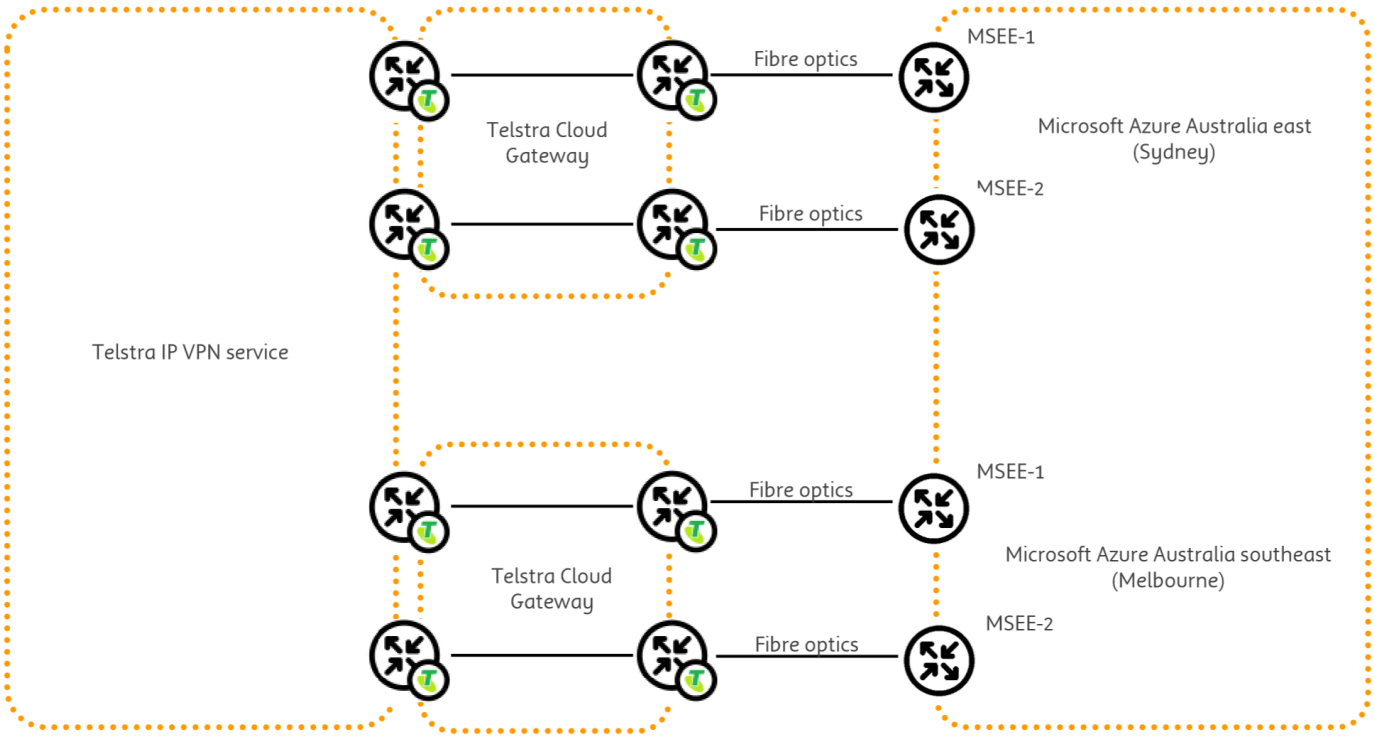

Figure 8 - Telstra Cloud Gateway georedundancy

5.1.2.2 Purpose

For Ambulance Victoria the purpose of requisitioning the Telstra Cloud Gateway is to connect directly to the Microsoft Azure Cloud via the 'ExpressRoute' service. This will allow private communication between Azure services and on-premises/Co-Lo networks. This is the recommended approach as it is more secure and transfer cost for egress data is cheaper than sending data straight out over the Internet. A direct connection is also potentially much faster than the alternative VPN service offered by Microsoft.

5.1.2.3 Sizing

The Telstra Cloud Gateway should be sized according to the number and size of links running through it. Initially this will be limited to a single 100mb ExpressRoute circuit so the Cloud Gateway should be the same size. The TCG product can easily be re-sized at a later date if required.

Key Design Decision
Ambulance Victoria would like to cater for regional redundancy so two Telstra Cloud Gateway services will be attached to the Ambulance Victoria MPLS network to enable redundancy for the connection of public cloud network private circuits. Initially this will be limited to dual 100mb ExpressRoute circuits and both the Telstra Cloud Gateways will be provisioned at 100mb. For future state these sizes may increase and Ambulance Victoria may also add a DirectConnect circuit into AWS.

Key Design Decision 5-2: Telstra Cloud Gateway

5.1.3 ExpressRoute Logical Connectivity

5.1.3.1 Overview

Microsoft Azure ExpressRoute allows an organisation to leverage Microsoft's cloud services through a dedicated private connection facilitated by an interconnect provider. ExpressRoute allows organisations to establish connections to three types of Microsoft services:

- Azure Private Services: Organisation's that utilise private-facing Azure services like Virtual Machines and Virtual Networks can address those private services through a private connection using private peering. Private

Peering also allows Azure services attached to a VNet to access on-premises private addresses. This is known as 'ExpressRoute Private Peering'.

- Azure Public Services: Organisations that utilise public-facing Azure services like Data Factory, Azure SQL, Azure Blobs or Azure public IP addresses can address those public services through a private connection using public peering. This is known as 'ExpressRoute Public Peering'.
- Office 365: Organisations that utilise Exchange Online, SharePoint Online, Yammer, and Skype for Business can address those public services through a private connection using public peering. This was known as 'ExpressRoute Microsoft Peering', it is now rolled into public peering as an option.

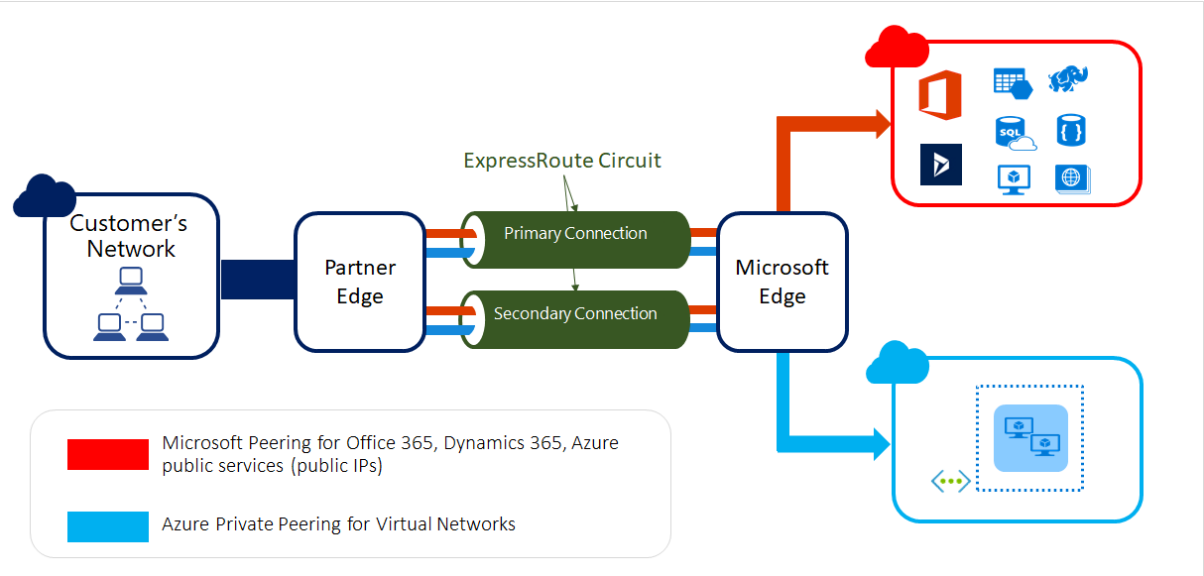

Figure 9 - ExpressRoute Overview

5.1.3.2 ExpressRoute Layout

There will be two ExpressRoute circuits deployed for Ambulance Victoria.

- One Telstra Cloud Gateway link which will terminate in the MPLS network. This link has an Australia Southeast (Melbourne) endpoint and will be connected to the shared services VNet in Australia Southeast.
    - The circuit will also be connected to a production shared services VNet in Australia East as the secondary route.
- One Telstra Cloud Gateway link which will terminate in the MPLS network. This link has an Azure Australia East (Sydney) endpoint and will be connected to the shared services VNet in Australia East.
    - The circuit will also be connected to a production shared services VNet in Australia Southeast as the secondary route.

5.1.3.3 Private Peering

All ExpressRoute circuits will have Private Peering enabled to allow internal traffic to connect directly to resources at either end.

5.1.3.4 Public Peering

Public Peering will be enabled on ExpressRoute circuits as Azure Site Recovery replicates virtual machines to an Azure storage endpoint. Public peering will enable the data to transmit across the ExpressRoute rather than across the internet.

5.1.3.5 Sizing

The ExpressRoute Circuit will be initially 100Mb in size. The relatively easy method of upgrading the link means Ambulance Victoria would prefer a smaller circuit to begin with and expand the bandwidth once utilisation demands as this will keep operational expenditure to a minimum.

5.1.3.6 ExpressRoute SKU

ExpressRoute circuits come in two SKUs; Standard and Premium. ExpressRoute premium is a collection of features listed below:

- Increased routing table limit from 4000 routes to 10,000 routes for private peering.
- Increased number of VNets that can be connected to the ExpressRoute circuit (default is 10).
- Global connectivity over the Microsoft core network. You will now be able to link a VNet in one geopolitical region with an ExpressRoute circuit in another region. Example: You can link a VNet created in Europe West to an ExpressRoute circuit created in Silicon Valley.
- Connectivity to Office 365 services and CRM Online.

Ambulance Victoria currently have no requirements for any of these additional features, the decision was made that features like Office 365 connectivity may be required in the future, so a Standard link will be provisioned initially. The link can be upgraded to Premium using a simple PowerShell command as required.

5.1.3.7 ExpressRoute Billing Options

There are two options for billing within ExpressRoute's; Metered and Unmetered. Metered connections are billed on the amount of egress traffic consumed whereas unmetered links are unlimited usage with a fixed cost. The break-even point for where an unmetered 50mbps connection becomes more expensive than a metered one is currently 11TB per month. Estimated Ambulance Victoria ExpressRoute usage is nowhere near that quantity, so the metered option will be deployed. This can be changed easily through the Azure portal later if network usage patterns within Ambulance Victoria change.

5.1.3.8 Network Resiliency

ExpressRoute circuits have inherent redundancy built in with two links inside the one ExpressRoute circuit. Ambulance Victoria have decided to plan on higher resiliency in a future by having two ExpressRoute links coming from different peering locations. Once usage patterns and requirements have been better identified through the usage of the Azure platform and additional link may be added.

Key Design Decision
The following design decisions were made for the Primary ExpressRoute circuit: • 100Mb circuit size • Metered billing option • Standard SKU selected • Private Peering enabled • Public Peering enabled • Peering Location will be Melbourne (secondary peering will be Sydney) The following design decisions were made for the Secondary ExpressRoute circuit: • 100Mb circuit size • Metered billing option • Standard SKU selected

- Private Peering enabled
- Public Peering enabled
- Peering Location will be Sydney (secondary peering will be Melbourne)

Key Design Decision 5-3: ExpressRoute

5.1.4 Internet Connectivity

5.1.4.1 IaaS Internet Traffic

Ambulance Victoria would like to treat Azure as an extension of their network (similar to the SDC-to-EHDC which has no firewalls between the datacentres). They would also like to ensure all internet connectivity for the IaaS environment is to be routed through an Azure based firewall infrastructure.

Key Design Decision
A DMZ will be established in both Azure regions to house Azure-based virtual firewall appliances. All external internet connectivity for the IaaS environment is to be routed through Azure based firewall. User Defined Routes will be implemented to force outbound traffic to the Azure firewall infrastructure.

Key Design Decision 5-4: IaaS Internet Traffic

5.1.4.2 PaaS Internet Traffic

Azure hosted application services (i.e. Web Apps) must utilise Azure internet connectivity as they are directly exposed to the internet with public IP addresses. This risk can be mitigated by utilise a Web Application Firewall (WAF)in front of the Web Apps and setting IIS controls to only allow access from the WAF

Key Design Decision
Azure PaaS services should utilise Azure internet connectivity and direct all traffic via a Web Application Firewall.

Key Design Decision 5-5: PaaS Internet Traffic

5.1.4.3 Global Load Balancing

The application layer approach to load balancing avoids the delays of DNS-based systems and can reroute traffic instantly. Utilising a DNS global load balancing service will enable failover between Azure regions.

5.1.4.3.1 Imperva Incapsula Global Server Load Balancing

Incapsula LBaaS[8] (Load Balancer-as-a-Service) is a cloud-based platform that distributes traffic across multiple data centres to improve performance and availability.

The service serves as a gateway for all incoming application layer traffic, which is then balanced among multiple data centres to ensure optimal performance. Rerouting occurs on the Incapsula service and is free of TTL-induced delays.

Real-time monitoring and control options provide a high degree of flexibility, including the option to set up geolocation-based rules to assist with compliance and other business goals.

Key Design Decision
The Imperva Incapsula Global Service Load Balancing will be licensed and deployed for global load balancing of all internet facing web services in Azure. Additional licenses will need to be purchased to enable the GSLB service on the Imperva service.

Key Design Decision 5-6: Imperva Incapsula Global Server Load Balancing

[8] https://www.incapsula.com/global-server-load-balancing.html

5.1.4.3.2 F5 BIG-IP DNS Service

F5 BIG-IP DNS[9] (formerly BIG-IP Global Traffic Manager) distributes DNS and user application requests based on business policies, data centre and cloud service conditions, user location, and application performance. The BIG-IP platform delivers F5's high-performance DNS services with visibility, reporting, and analysis; hyperscales and secures DNS responses geographically to survive DDoS attacks; delivers a complete, real-time DNSSEC solution; and ensures high availability of global applications in all hybrid environments.

Ambulance Victoria currently have F5's deployed in their datacentres and are licensed for the full F5 capability. Due to the Private IaaS pattern having only internal network connectivity, the Imperva cloud service cannot perform regional load balancing for the internal networks.

Key Design Decision
The F5 BIG-IP DNS service will be deployed for global load balancing of the "Private IaaS Legacy VM HA Pattern" as per the "Azure Application Deployment Patterns Reference Architecture" document.

Key Design Decision 5-7: F5 BIG-IP DNS Service

5.1.4.4 Firewalls

5.1.4.4.1 Overview

In computing, a firewall is a network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules. A firewall typically establishes a barrier between a trusted internal network and untrusted external network, such as the Internet.

[9] https://www.f5.com/pdf/products/big-ip-dns-datasheet.pdf

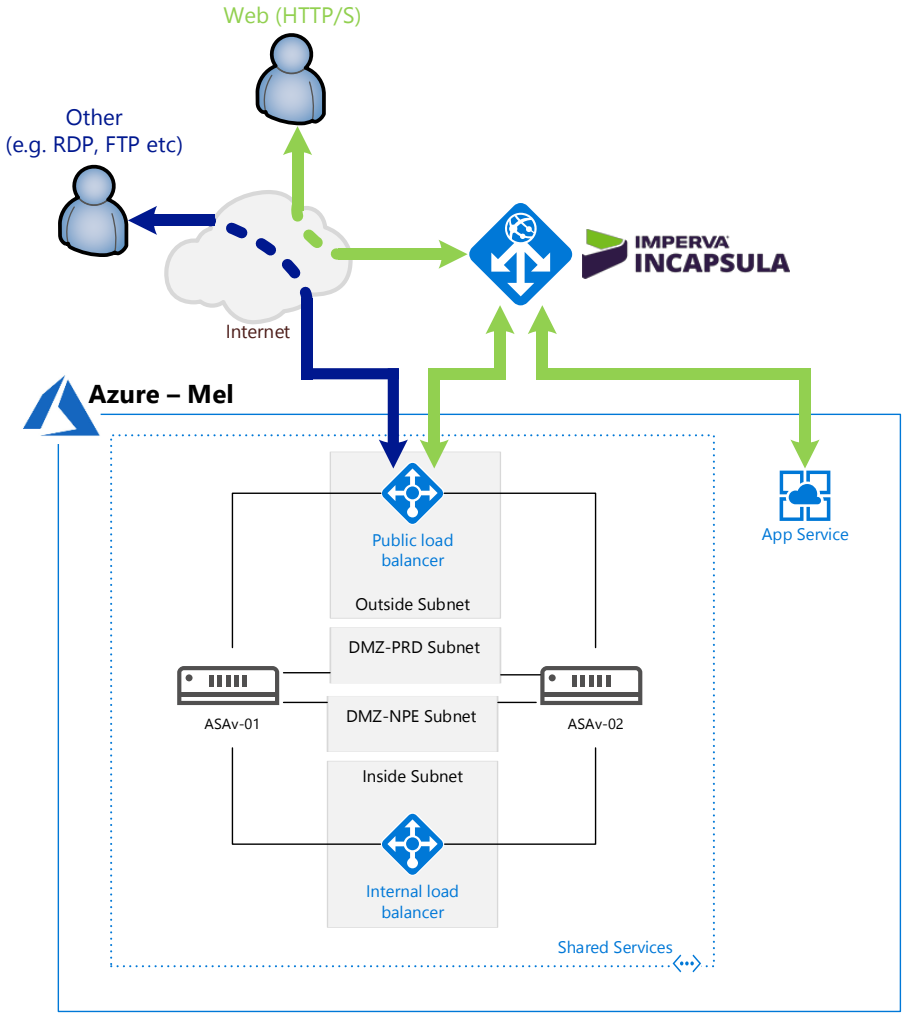

Figure 10 - Network Firewall Reference Architecture

5.1.4.4.2 Imperva Incapsula WAF

A web application firewall (or WAF) filters, monitors, and blocks HTTP traffic to and from a web application. A WAF is differentiated from a regular firewall in that a WAF is able to filter the content of specific web applications while regular firewalls serve as a safety gate between servers. By inspecting HTTP traffic, it can prevent attacks stemming from web application security flaws, such as SQL injection, cross-site scripting (XSS), file inclusion, and security misconfigurations.

Ambulance Victoria currently have the Incapsula cloud-based web application firewall[10] (WAF) which is a managed service that protects from application layer attacks, including all OWASP top 10 and even zero-day threats.

The service is PCI-certified, highly customizable, SIEM ready and fine-tuned for blocking threats with minimal false positives.

Ambulance Victoria have decided this product is the enterprise architecture model to use for protecting cloud hosted web services. The product is currently licensed for 25 applications with 21 free licenses.

The Incapsula WAF acts as a gateway for all incoming traffic to your web application. This puts it in a perfect position to filter out malicious visitors and requests like SQL injections and XSS attacks among others.

Key Design Decision

[10] https://www.incapsula.com/website-security/web-application-firewall.html

The Incapsula WAF will be utilised for all public facing web services within Azure. For Public IaaS pattern the Azure based virtual firewall appliance will be configured to only allow connections from the Incapsula cloud service[11]. For Public PaaS pattern the WebApp configuration will be configured to only allow connections from the Incapsula cloud service.

Key Design Decision 5-8: Imperva Incapsula WAF

5.1.4.4.3 Cisco Adaptive Security Appliance virtual (ASAv) Firewall

ASAv is the virtualized version of Cisco's best-selling Adaptive Security Appliance (ASA).

The physical Cisco ASA and Cisco ASAv support the same rich policy constructs. Virtual and physical domains are coalesced into a single policy domain so the same policies can be applied to all Cisco ASAs, whether they are physical or virtual.

Cisco ASAv offers the same features as a physical Cisco ASA, including VPN services that can be deployed in the virtual domain. Site-to-site, remote-access, and clientless VPN services can be deployed quickly in a private cloud or over a virtual infrastructure in response to demand.

Cisco ASAv offers the REST API, an HTTP-based interface that facilitates management of the appliance, including changing the security policy and monitoring the status. Using REST APIs, multiple cloud management solutions can be used to manage both physical and virtual instances of Cisco ASA.

- FREE TRIAL- ASAv has a demo mode that runs with reduced performance. No license required.
- Supported Azure Instances: Standard_D3 and Standard_D3_V2
- ASAv is integrated with Azure Security Center
- ASAv is available in the Azure Government Cloud.

Ambulance Victoria currently use Cisco ASA appliances in both datacentres to perform packet filtering firewall services for the inbound internet connectivity and would like to align toolsets to have a “single pane of glass” management interface where possible.

[11] https://docs.microsoft.com/en-us/azure/app-service/app-service-ip-restrictions

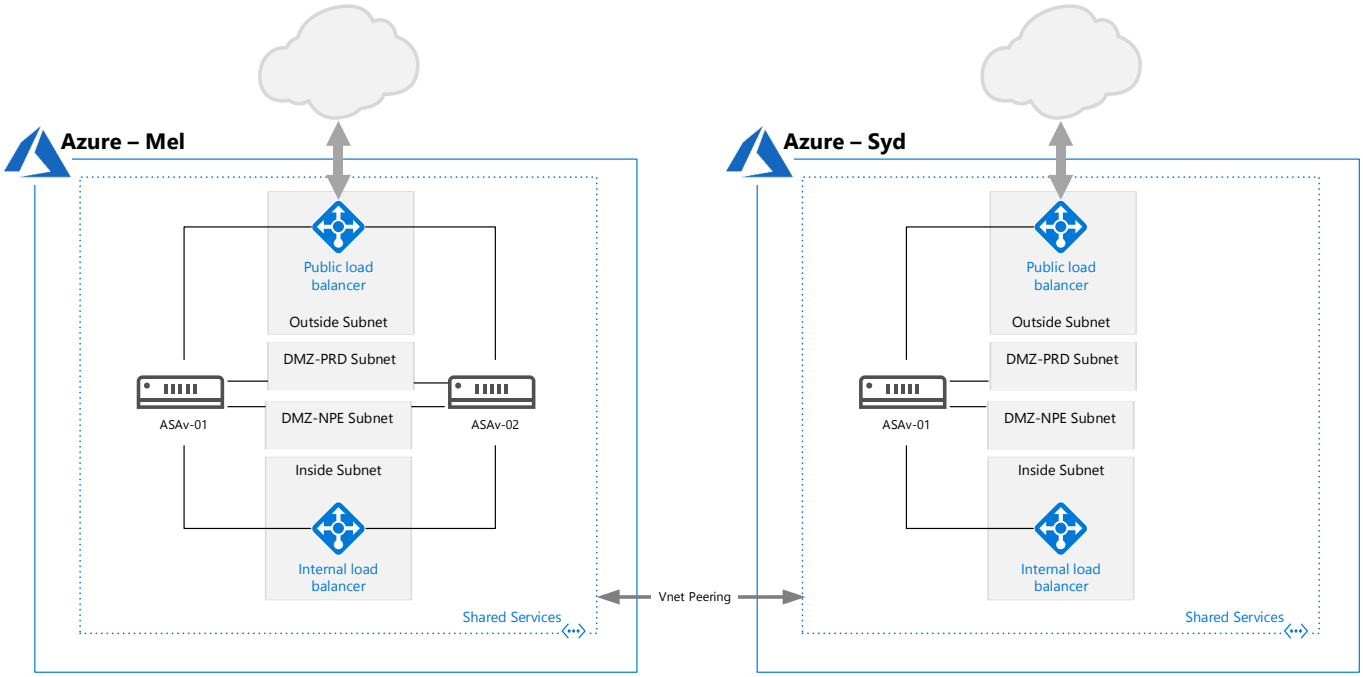

Figure 11 - ASAv Reference Architecture

Key Design Decision
Dual Cisco ASAv's will be deployed into the Azure VNet environment and will be utilised for all public facing services (Public IaaS & Public SaaS patterns) within Azure, except for Public PaaS, which will only use the WAF since the Public PaaS pattern is directly connected to the internet and is not deployed in the VNet. For the Sydney region only one ASAv will be deployed. For Public IaaS patterns the Azure based virtual firewall appliance will be configured to only allow connections from the Incapsula cloud service.

Key Design Decision 5-9: Cisco Adaptive Security Appliance virtual (ASAv) Firewall

5.2 Azure Internal Network Design

5.2.1 Overview

This section covers the networking components within the Azure service. Below is the overall Azure internal networking design diagram:

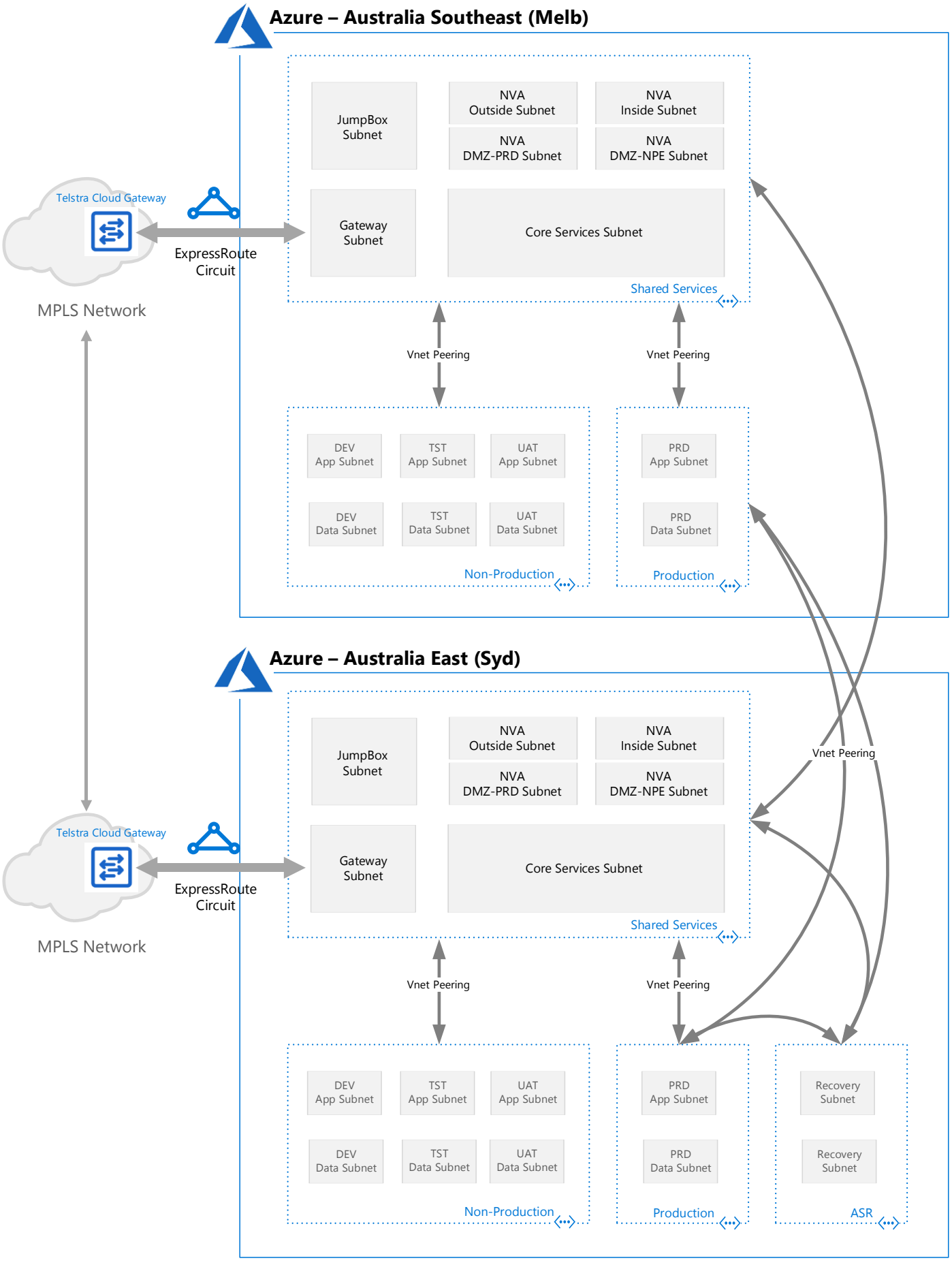

Figure 12 - Azure Internal Network Diagram

5.2.2 Virtual Networks

Azure networking is delivered through use of Virtual Networks (VNets). VNets allow for the connection of IaaS and some PaaS services within Azure, including the definition of DHCP address blocks, DNS settings, security policies, and routing.

Key Design Decision
Ambulance Victoria have defined their workloads into 4 main environments, Shared, Production and Non-Production and Azure Site Recovery. Shared consists of core operational components which are consumed across multiple environments, such as domain controllers, monitoring services, firewalls, build agents, jump boxes etc. Production consists of all Ambulance Victoria production workloads while Non-Production consists of all non-production workloads. Virtual networks will be created in both Azure regions as per the following: • Shared • Production • Non-Production The following virtual network will only be created in the Australia East (Sydney) region as it is just for on-premise DR purposes: • ASR

Key Design Decision 5-10: Virtual Networks

5.2.3 Virtual Network Peering

Virtual network peering enables you to seamlessly connect two Azure virtual networks. This enables resources in different virtual networks to communicate with each other with the same bandwidth and latency as though the resources were in the same virtual network. The traffic between virtual machines in the peered virtual networks is routed through the Microsoft backbone infrastructure, much like traffic is routed between virtual machines in the same virtual network, through private IP addresses only.[12]

Key Design Decision
The Production and Non-Production VNets will be peered with the Shared VNet which will enable them to communicate with the on-premises network as well as any other services hosted in the Shared network. This configuration also ensures the Production and Non-Production VNets cannot communicate with each other. The Melbourne and Sydney Shared VNets will be peered with each other to allow core services to communicate with each other e.g. AD replication between the Active Directory domain controllers in each region. The Melbourne and Sydney Production VNets will be peered with each other to allow application communication between regions. The ASR VNet in Sydney will be peered with the Shared VNet in the Sydney region and the Production VNets in both regions.

Key Design Decision 5-11: Virtual Network Peering

5.2.4 Virtual Network Gateway

A virtual network gateway is composed of two or more virtual machines that are deployed to a specific subnet called the GatewaySubnet. The VMs are created in the GatewaySubnet when you create the virtual network gateway. Virtual network gateway VMs are configured to contain routing tables and gateway services specific to the gateway. You can't

[12] https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-peering-overview

directly configure the VMs that are part of the virtual network gateway and you should never deploy additional resources to the GatewaySubnet.[13]

Key Design Decision
A virtual network gateway is required to be deployed in both the Shared virtual networks as they will act as endpoints for ExpressRoute and allow virtual networks to be peered.

Key Design Decision 5-12: Virtual Network Gateway

5.2.5 Subnets

Azure VNets can be logically separated into IPv4 subnets. All subnets within a VNet can communicate unrestricted to each other by default and Azure provides two controls for networking separation and isolation, Network Security Groups (NSGs) and User Defined Routes (UDRs).

Key Design Decision
To align with a more mature, layered security approach while providing flexibility and scalability the virtual networks will contain the following subnets: Shared VNet • NVA-Outside – Hosts the public load balancer for the firewalls • NVA-DMZ-Prod – Hosts public facing production servers • NVA-DMZ-NPE – Hosts public facing non-production servers • NVA-Inside – Hosts internal load balancer for the firewalls • CoreServices – Core operational components such as domain controllers, identity services, etc. • JumpBox – Isolated locked down area dedicated for servers utilised to remotely administer infrastructure and Azure resources. • GatewaySubnet – Dedicated area required for Azure VPN gateways to function. Production VNet To provide segregation and security between application and data workloads, the production virtual network will contain the following subnets: • App – Production application workloads. This can include web servers, application servers or a combination of both. This provides more flexibility than having separate web and app subnets as sometimes applications do not segregate their web server from their application engine. • Data – Production database workloads. Non-Production VNet Ambulance Victoria have three types of Non-Production workloads, Development, Test and UAT. To cater for all Non-Production workloads the Non-Production virtual network will contain the following subnets: • DEV-App – Development web/application workloads. • DEV-Data – Development database workloads. • TST-App – Test web/application workloads. • TST-Data – Test database workloads. • UAT-App – UAT web/application workloads. • UAT-Data – UAT database workloads.

[13] https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpngateways

ASR VNet

Ambulance Victoria have a requirement to replicate some Platinum on-premise servers into Azure to enable site recovery in the event of on-premise virtual machine failures. No subnets are to be segmented out of the ASR supernet at this stage.

Key Design Decision 5-13: Subnets

5.2.5.1 Subnet Sizing

Subnets are sized based on several factors:

- Potential growth for resources on that subnet
- Best practice for subnet size

Key Design Decision

The following subnets have been allocated by Ambulance Victoria internal network provider.

VNet	Addresses each site	Azure Melbourne	Azure Sydney
Shared VNet	1022	172.24.244.0/22	172.24.224.0/22
nva-outside-subnet	30	172.24.243.64/27	172.24.227.64/27
nva-dmz-prd-subnet	254	172.24.240.0/24	172.24.224.0/24
nva-dmz-npd-subnet	254	172.24.241.0/24	172.24.225.0/24
nva-inside-subnet	30	172.24.243.96/27	172.24.227.96/27
core-services-subnet	254	172.24.242.0/24	172.24.226.0/24
jumpbox-subnet	30	172.24.243.0/27	172.24.227.0/27
GatewaySubnet	30	172.24.243.32/27	172.24.227.32/27
Production VNet	1022	172.24.244.0/22	172.24.228.0/22
prd-app-subnet	254	172.24.244.0/24	172.24.228.0/24
prd-data-subnet	254	172.24.245.0/24	172.24.229.0/24
Non-production VNet	2046	172.24.248.0/21	172.24.232.0/22
dev-app-subnet	126	172.24.252.0/25	172.24.235.0/25
dev-data-subnet	126	172.24.253.0/25	172.24.235.128/25
tst-app-subnet	254	172.24.250.0/24	172.24.234.0/25
tst-data-subnet	254	172.24.251.0/24	172.24.234.128/25
uat-app-subnet	254	172.24.248.0/24	172.24.232.0/24
uat-data-subnet	254	172.24.249.0/24	172.24.233.0/24
ASR	510	-	172.24.236.0/23
Recovery-Subnet-01	510	-	172.24.236.0/23

Key Design Decision 5-14: Subnet Sizing

5.2.6 Network Security Groups

A network security group (NSG) contains a list of access control list (ACL) rules that allow or deny network traffic to your VM instances in a Virtual Network. NSGs can be associated with either subnets or individual VM instances within that

subnet. When an NSG is associated with a subnet, the ACL rules apply to all the VM instances in that subnet. In addition, traffic to an individual VM can be restricted further by associating an NSG directly to that VM[14].

Key Design Decision
NSGs will not be deployed as the network zoning model determines network layer security is not required for internal communication. In the future if further network layer controls are required then NSGs can be deployed.

Key Design Decision 5-15: Network Security Groups

5.2.7 Virtual Network Service Endpoints

Virtual Network (VNet) service endpoints[15] extend your virtual network private address space and the identity of your VNet to the Azure services, over a direct connection. Endpoints allow you to secure your critical Azure service resources to only your virtual networks. Traffic from your VNet to the Azure service always remains on the Microsoft Azure backbone network.

Key Design Decision
VNet Endpoints will be deployed for any eligible Azure services that need to be consumed within the legacy applications i.e. applications using virtual networks and virtual machines i.e. not PaaS

Key Design Decision 5-16: Virtual Network Service Endpoints

5.2.8 Load Balancers

5.2.8.1 Internal Load Balancer

Azure Internal Load Balancer[16] (ILB) provides load balancing between resources that reside inside a cloud service or a virtual network within a regional scope. The Azure infrastructure restricts access to the load-balanced virtual IP (VIP) addresses of a cloud service or to a virtual network and are never directly exposed to an internet endpoint. Internal line-of-business applications that run in Azure and are accessed from within Azure or from on-premises resources.

Key Design Decision
Internal Load Balancers will be deployed to load balance internal Azure traffic between the firewalls in the Shared virtual networks. Further ILBs may be required future state for other Azure resources.

Key Design Decision 5-17: Internal Load Balancer

5.2.8.2 Public Load Balancer

Azure Public Load Balancer provides a public IP address and port number for load balancing internet facing traffic to the private IP address and port number of a virtual machine or service. Load balancing rules allow you to distribute specific types of traffic between multiple virtual machines or services. For example, you can spread the load of web request traffic across multiple web servers or web roles.

Key Design Decision

[14] https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-nsg
[15] https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-service-endpoints-overview
[16] https://docs.microsoft.com/en-us/azure/load-balancer/load-balancer-internal-overview

Public Load Balancers will be deployed to load balance inbound internet traffic between the firewalls in the Shared virtual networks. No other public load balancers will be deployed as all inbound internet traffic must route via the firewalls.

Key Design Decision 5-18: Public Load Balancer

5.2.9 DNS Services

The Domain Name System, or DNS, is responsible for translating (or resolving) a service name to its IP address. Within Azure, DNS can be configured at the virtual network level which feeds the correct DNS entries down to the subnets and subsequently to the virtual machines and web applications. When configuring DNS in Azure you can utilise Azure DNS to provide name resolution between virtual machines located in the same cloud service or virtual network or use your own custom DNS server/s for name resolution.

Key Design Decision
The DNS role will be deployed onto the Active Directory Domain Controllers deployed in the Shared Service VNet. DNS will be configured at the VNet level using custom AD DNS server services to be deployed in Azure to allow name resolution of Active Directory domain requests. All VNets will be configured to use the production DNS servers. A tertiary on-premises DNS server will also be added to the list to provide additional redundancy. Azure Public PaaS web apps will utilize the Azure DNS service and not the Ambulance Victoria DNS servers. DMZ servers will utilize local host files and external DNS servers as selected by Ambulance Victoria.

Key Design Decision 5-19: DNS Services

5.2.10 User-Defined Routes

User-Defined Routes (UDR) provide you a way to override Azure's default system routes, or to add additional routes to a subnet's route table. In Azure, you create a route table, then associate the route table to one or more virtual network subnets. Each subnet can have one route table associated to it and once associated, the routes within it are combined with, or override, the default routes Azure adds to a subnet by default.[17]

Key Design Decision
A route table will be configured on all subnets to route all traffic destined for either the DMZ or the internet to the firewall's internal load balancer IP address. Firewall: Multiple route tables will be configured for both the production and non-production the firewalls to route traffic from the DMZ-PRD and DMZ-NPE firewall subnets to the Outside firewall subnets as per application design and for internet traffic to flow correctly

Key Design Decision 5-20: User-Defined Routes

[17] https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-udr-overview#custom-routes

6 Azure Platform Components

6.1 Storage

6.1.1 Storage Accounts

Microsoft Azure Storage is a Microsoft-managed cloud service that provides storage that is highly available, secure, durable, scalable, and redundant. Storage Accounts come in two forms: General Purpose and Blob Storage.

A General-Purpose storage account gives you access to Azure Storage services such as Tables, Queues, Files, Blobs and Azure virtual machine disks. These storage services are included in every storage account created. A storage account provides a unique namespace for working with the blobs, queues, and tables. This type of storage account has two performance tiers:

- A standard storage performance tier which allows you to store Tables, Queues, Files, Blobs and Azure virtual machine disks. This tier is hosted on Hard Disk Drives (HDD)
- A premium storage performance tier which currently only supports Azure virtual machine disks. This tier is hosted on Solid State Drives (SSD).

A Blob Storage Account is a specialized storage account for storing unstructured data as blobs (objects) in Azure Storage. Blob storage accounts are similar to the general-purpose storage accounts and share all the durability, availability, scalability, and performance features and API consistency for block blobs and append blobs. *For applications requiring only block or append blob storage, it is recommended to use Blob storage accounts.*

This type of storage account has two tiers:

- A Hot access tier which indicates that the objects in the storage account will be more frequently accessed. This allows you to store data at a lower access cost.
- A Cool access tier which indicates that the objects in the storage account will be less frequently accessed. This allows you to store data at a lower data storage cost.

Both storage accounts are available with four redundancy types:

- Locally redundant storage (LRS) is replicated three times within a single facility in a single region. LRS protects your data from normal hardware failures, but not from the failure of a single facility. Premium Storage accounts are only available with LRS.
- Zone-redundant storage (ZRS) is replicated three times across two to three facilities, either within a single region or across two regions, providing higher durability than LRS. ZRS ensures that your data is durable within a single region.
- Geo-redundant storage (GRS) is enabled for your Standard storage account by default when you create it. With GRS, your data is replicated three times within the primary region and three times in a secondary region. In the event of a failure at the primary region, Azure Storage will fail over to the secondary region. It is important to note that GRS does not fail over to the secondary location unless there is TOTAL data centre outage
- Read-access geo-redundant storage (RA-GRS) allows you to have higher read availability for your storage account by providing Read-only access to the data replicated to the secondary location. When you enable this feature, the secondary location can be used to achieve higher availability in the event the data is not available in the primary region. Read-access geo-redundant storage is recommended for maximum availability and durability.

Key Design Decision

General Purpose, Hot Access, LRS only Storage accounts will be deployed to host the following:

- Diagnostic data of virtual machines. (one each per environment, e.g development, test, UAT, production).
- Install files and scripts for deploying VMs (one shared across all environments).
- Recovery Vault in Australia Southeast that is GRS enabled to allow for potential Microsoft initiated failover of the recovery vault to the Sydney region.
- Any storage accounts created as part of an Azure App Service deployment will be hosted in the App Service's own resource group.

Key Design Decision 6-1: Storage Accounts

6.1.2 Storage Account Access Control

Access to the storage accounts can be controlled using one or more of the following methods

a) Role-Based Access Control (RBAC) – RBAC can be used to control access to storage accounts. Some of the key roles built-in roles storage accounts are listed below.
   - Owner – can manage everything, including access.
   - Contributor – can do anything the owner can do except assign access. Someone with this role can view and regenerate the storage account keys. With the storage account keys, they can access the data objects.
   - Reader – can view information about the storage account, except secrets. For example, if you assign a role with reader permissions on the storage account to someone, they can view the properties of the storage account, but they can't make any changes to the properties or view the storage account keys.
   - Storage Account Contributor – can manage the storage account – they can read the subscription's resource groups and resources and create and manage subscription resource group deployments. They can also access the storage account keys, which in turn means they can access the data plane.
   - User Access Administrator – can manage user access to the storage account. For example, they can grant Reader access to a specific user.
   - Virtual Machine Contributor – can manage virtual machines but not the storage account to which they are connected. This role can list the storage account keys, which means that the user to whom you assign this role can update the data plane.

b) Storage Account access keys - Azure storage accounts can be accessed from an application by providing an account name and access key in the connection string. Storage account keys are 512-bit strings created by Azure that, along with the storage account name, can be used to access the data objects stored in the storage account, e.g. blobs, entities within a table, queue messages, and files on an Azure File share. Controlling access to the storage account keys controls access to the data plane for that storage account. Each storage account has two keys referred to as "Key 1" and "Key 2" in the Azure portal and in the PowerShell cmdlets. These can be regenerated manually using one of several methods, including, but not limited to using the Azure portal, PowerShell, the Azure CLI, or programmatically using the .NET Storage Client Library or the Azure Storage Services REST API.

c) Shared Access Signatures (SAS) - A shared access signature (SAS) is a URI that grants restricted access rights to Azure Storage resources. A SAS provides delegated access to resources in your storage account. With a SAS, you can grant applications access to resources in the storage account, without sharing your account keys, thereby restricting administrative rights on the storage account and limiting the risk of malicious or negligent use. The following options can be set for access using SAS:
   - Allowed services: Blob, File, Queue, Table
   - Allowed resource types: Service, Container, Object.
   - Allowed permissions: Read, Write, Delete, List, Add, Create, Update.
   - Process: Start, expiry date/time and time zone.
   - Allowed IP addresses: Access restricted based on source IP addresses.

- **Allowed protocols:** HTTPS only (default), HTTPS and HTTP (not recommended)
- **Signing key:** key1 or key2

Key Design Decision
Ambulance Victoria have a requirement for the systems engineering team to be administrators of all storage accounts. The systems engineering team will be granted OWNER rights to each storage account created.

Key Design Decision 6-2: Storage Account Access Control

6.1.3 Storage Account Encryption

There are three forms of encryption available for the Storage services.

Encryption at rest

You can enable Storage Service Encryption (SSE) on either the Files service (preview) or the Blob service for an Azure storage account. If enabled, all data written to the specific service is encrypted before it is written. When you read the data, it is decrypted before it is returned.

Client-side encryption

The storage client libraries have methods you can call to programmatically encrypt data before sending it across the wire from the client to Azure. It is stored encrypted, which means it also is encrypted at rest. When reading the data back, you decrypt the information after receiving it.

Encryption in transit

All access to storage accounts is can be done via HTTPS with SAS keys for authentication.

Key Design Decision
The encryption in transit (HTTPS) is always enabled and the SSE at rest encryption will be enabled by default on all storage accounts to comply with Ambulance Victoria security policy. Client-side encryption maybe used by other not yet deployed applications.

Key Design Decision 6-3: Storage Account Encryption

6.1.4 Managed Disks

Azure Managed Disks is an abstraction of current Standard and Premium storage disk in Azure Storage. Using Managed Disks, Ambulance Victoria will only need to specify:

- Storage type (Standard or Premium)
- Size of disk needed
- Azure region

Based on this information, Azure will create and manage the disk accordingly. There are 4 key value propositions of Azure Managed Disks:

1. **Management**: Managed Disks will be available as a top-level resource (like VMs). Ambulance Victoria do not have to worry about creating and managing Storage Accounts. Managed Snapshots and Images are also top-level resources, which will enable Ambulance Victoria to take point-in-time back-ups of their Managed Disks and VM configuration, enabling the creation of Disks and/or VMs at a later point.

2. Scale: Managed Disks abstract the concept of storage accounts, removing the need to manage storage accounts. This enables Ambulance Victoria to scale regardless of the limitations associated with the storage accounts, such as 40 disks per storage account and 100 storage accounts per subscription.
3. Availability: This is the primary focus for Managed Disks. To enable higher availability, we enforce two constraints for Availability Set using Managed Disks:
    - Storage Isolation of VM Availability Set.
    - VM fault domains align with Managed Disk fault domains.

Security: Managed Disks will be exposed as the logical disk and removes the public internet-facing endpoint to the storage account hosting VM disk data. Ambulance Victoria will also be able to define RBAC policies for their Disks, like Virtual Machines. Another security feature is encryption of data at rest. Managed Disks are encrypted using SSE (Storage Service Encryption) by default. All data are encrypted using AES 256-bit encryption. The keys for encryption are managed by Microsoft.

Managed Disk names are unique within the resource group. A disk can be created/deleted and attached/detached/reattached to a VM as required. It is also possible to create Availability Sets for VMs using Managed Disks. For such Availability Sets, Managed Disks service will enforce the Compute and Storage fault domain alignment, offering higher availability than is possible with the Storage Account deployment model.

A point-in-time backup of the Managed Disk can be taken as a Snapshot that can then be used to create Images from the VM or from Snapshots.

Key Design Decision
Managed disks are to be used for most VM workloads using standard type disks. Sizing will be minimum recommended for the deployment type since managed disks are priced based on allocated storage rather than utilised storage. Managed disks are encrypted by default with Microsoft managing the encryption and key, Ambulance Victoria will make use of this technology as the excessive administrative overhead of managing their own keys is currently not possible due to resourcing restraints.

Key Design Decision 6-4: Managed Disks

6.2 Monitoring and Alerts

Ambulance Victoria are currently undergoing a strategic review of their event management capability.

Key Design Decision
No monitoring services will be deployed at this time as per request by Ambulance Victoria. The Azure environment will need to be reviewed for alignment to the enterprise monitoring strategy once it is established.

Key Design Decision 6-5 - Monitoring and Alerts

6.3 Backup and Recovery

Ambulance Victoria have decided to utilise Azure backup and recovery features for Azure workloads rather than send the backups back on-premise to the corporate backup solution.

For those resources being backed up, the initial backup policies will be:

Key Design Decision

All backup schedules and retention policies should align to the existing backup policies as determined by Ambulance Victoria.

6.3.1 Commvault

Commvault[18] software is an enterprise-level data platform that contains modules to back up, restore, archive, replicate, and search data. It is built from the ground-up on a single platform and unified code base.

Data is protected by installing agent software on the physical or virtual hosts, which use operating system or application native APIs to protect data in a consistent state. Production data is processed by the agent software on client computers and backed up through a data manager, the MediaAgent, to disk, tape, or cloud storage. All data management activity in the environment is tracked by a centralized server, the CommServe, and can be managed by administrators through a central user interface. End users can access protected data using web browsers and mobile devices.

Key Design Decision
Commvault should be configured to back up the Azure instances and store the backup in the Azure cloud. Agents will be pushed to all Azure VMs. Where possible API integration between Commvault and Azure should be configured.

Key Design Decision 6-6: Commvault

6.3.2 Azure Backup

Azure Backup[19] is the Azure-based service you can use to back up (or protect) and restore your data in the Microsoft cloud. Azure Backup replaces your existing on-premises or off-site backup solution with a cloud-based solution that is reliable, secure, and cost-competitive. Azure Backup offers multiple components that you download and deploy on the appropriate computer, server, or in the cloud. The component, or agent, that you deploy depends on what you want to protect. All Azure Backup components (no matter whether you're protecting data on-premises or in the cloud) can be used to back up data to a Recovery Services vault in Azure

Azure IaaS VM Backup which is part of Azure Fabric is the mechanism for backing up Azure based virtual machines to a Recovery Service Vault.

Recovery Services Vault

A Recovery Services vault is a storage entity in Azure that houses data. The data is typically copies of data, or configuration information for virtual machines (VMs), workloads, servers, or workstations. You can use Recovery Services vaults to hold backup data for various Azure services such as IaaS VMs (Linux or Windows) and Azure SQL databases. Recovery Services vaults support System Center DPM, Windows Server, Azure Backup Server, and more. Recovery Services vaults make it easy to organize your backup data, while minimizing management overhead[20].

Recommendation
Recovery Services vaults should be deployed to back up the VMs if Commvault cloud requirements cannot be met.

Recommendation 6-1: Azure Backup

[18] https://www.commvault.com/
[19] https://docs.microsoft.com/en-au/azure/backup/backup-azure-vms-introduction
[20] https://docs.microsoft.com/en-us/azure/backup/backup-azure-recovery-services-vault-overview

6.3.3 Azure Site Recovery Service

Azure Site Recovery[21] (ASR) helps ensure business continuity by keeping business apps and workloads running during outages. Site Recovery replicates workloads running on physical and virtual machines (VMs) from a primary site to a secondary location. When an outage occurs at your primary site, you fail over to secondary location, and access apps from there. After the primary location is running again, you can fail back to it.

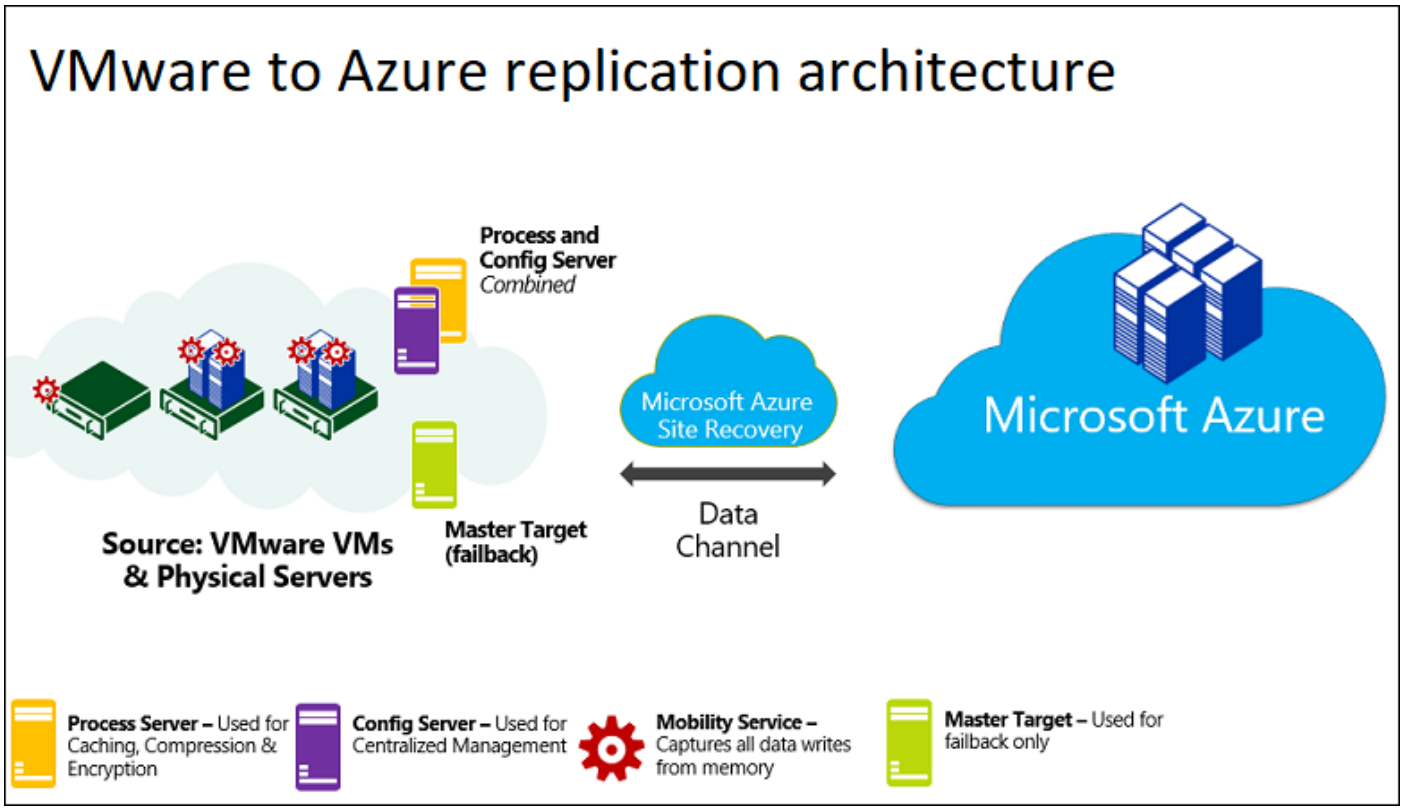

Figure 13 - ASR Architectural Diagram

Architectural Components

Component	Requirement	Details
Azure	An Azure subscription, Azure Storage account, and Azure network.	Replicated data from on-premises VMs is stored in the storage account. Azure VMs are created with the replicated data when you run a failover from on-premises to Azure. The Azure VMs connect to the Azure virtual network when they're created.
ExpressRoute Public peering[22]	ExpressRoute enabled for Public peering for Site Recovery replication.	Site Recovery replicates data to an Azure Storage account over a public endpoint. To use ExpressRoute for Site Recovery replication, you can utilize public peering or Microsoft peering. Microsoft peering is the recommended routing domain for replication. After virtual machines or servers fail over to an Azure virtual network, you can access them using private peering. Replication is not supported over private peering.

[21] https://docs.microsoft.com/en-au/azure/site-recovery/site-recovery-overview
[22] https://docs.microsoft.com/en-us/azure/site-recovery/concepts-expressroute-with-site-recovery#on-premises-to-azure-replication-with-expressroute

Configuration server machine	A single on-premises machine. We recommend that you run it as a VMware VM that can be deployed from a downloaded OVF template. The machine runs all on-premises Site Recovery components, which include the configuration server, process server, and master target server.	Configuration server: Coordinates communications between on-premises and Azure, and manages data replication. Process server: Installed by default on the configuration server. It receives replication data; optimizes it with caching, compression, and encryption; and sends it to Azure Storage. The process server also installs Azure Site Recovery Mobility Service on VMs you want to replicate, and performs automatic discovery of on-premises machines. As your deployment grows, you can add additional, separate process servers to handle larger volumes of replication traffic. Master target server: Installed by default on the configuration server. It handles replication data during failback from Azure. For large deployments, you can add an additional, separate master target server for failback.
VMware servers	VMware VMs are hosted on on-premises vSphere ESXi servers. We recommend a vCenter server to manage the hosts.	During Site Recovery deployment, you add VMware servers to the Recovery Services vault.
Replicated machines	Mobility Service is installed on each VMware VM that you replicate.	We recommend that you allow automatic installation from the process server. Alternatively, you can install the service manually or use an automated deployment method, such as System Center Configuration Manager.

Figure 14 - ASR Architectural Components[23]

Key Design Decision
Azure Site Recovery Service (ASR) will be deployed to replicate eligible on-premise servers to Azure and enable failover and failback of the servers. It will be hosted in the Australia East (Sydney) region as it is purely for DR.

Key Design Decision 6-7: Azure Site Recovery Service

6.3.3.1 Site Recovery Deployment Planner

Site Recovery Deployment Planner[24] is a command-line tool for both Hyper-V to Azure and VMware to Azure disaster recovery scenarios. You can remotely profile your VMware VMs by using this tool (with no production impact whatsoever) to understand the bandwidth and storage requirements for successful replication and test failover. You can run the tool without installing any Site Recovery components on-premises. To get accurate achieved throughput results, run the planner on a Windows Server that meets the minimum requirements of the Site Recovery configuration server that you eventually need to deploy as one of the first steps in production deployment.

Key Design Decision

[23] https://docs.microsoft.com/en-us/azure/site-recovery/vmware-azure-architecture
[24] https://docs.microsoft.com/en-us/azure/site-recovery/site-recovery-deployment-planner

Site Recovery Deployment Planner will be deployed to determine the capacity requirements for the ASR service.

Key Design Decision 6-8: Site Recovery Deployment Planner

6.3.3.2 ASR Resource Group

Key Design Decision
A separate ASR resource group will be created in the Australia East region to contain all the resources required for the Azure Site Recovery service.

Key Design Decision 6-9: ASR Resource Group

6.3.3.3 Storage Account

Images of replicated machines are held in Azure storage. Azure VMs are created from the storage when you fail over from on-premises to Azure. The storage account must be in the same region as the Recovery Services vault.

Key Design Decision
A single general purpose v1 read-access geo-redundant storage account will be deployed in the Australia East region and registered into the ASR resource group

Key Design Decision 6-10: Storage Account

6.3.3.4 Recovery Service Vault for ASR

A vault holds metadata and configuration information for VMs, and other replication components.

Key Design Decision
A single general Recovery Service Vault will be deployed in the Australia East region and registered into the ASR resource group.

Key Design Decision 6-11: Recovery Service Vault for ASR

6.3.3.5 Azure Recovery Service Network Configuration

Azure Site Recovery replicated VMs are deployed into a recovery subnet when a failover is performed.

Key Design Decision
All replicated virtual machines will be deployed into the ASR virtual network in the Australia East region.

Key Design Decision 6-12: Azure Recovery Service Network Configuration

6.3.3.6 On-premise Configuration Server

You deploy an on-premises configuration server[25] when you use Azure Site Recovery for disaster recovery of VMware VMs and physical servers to Azure. The configuration server coordinates communications between on-premises VMware and Azure. It also manages data replication.

Configuration Server Sizing

CPU	Memory	Cache disk size	Data change rate	Protected machines
8 vCPUs (2 sockets * 4 cores @ 2.5 GHz)	16 GB	300 GB	500 GB or less	Replicate fewer than 100 machines.
12 vCPUs (2 sockets * 6 cores @ 2.5 GHz)	18 GB	600 GB	500 GB to 1 TB	Replicate 100-150 machines.

[25] https://docs.microsoft.com/en-us/azure/site-recovery/vmware-azure-deploy-configuration-server

16 vCPUs (2 sockets * 8 cores @ 2.5 GHz)	32 GB	1 TB	1 TB to 2 TB	Replicate 150-200 machines.

Figure 15 - Configuration Server Sizing Recommendations

Configuration Server Network Settings

The configuration server requires network communication between itself-to-Azure and itself-to-VMware.

Key Design Decision

A single configuration server using the OVA (Open Virtualization Application) template will be deployed into the State Datacenter (SDC) environment.

The configuration server initial sizing will be:

CPU	Memory	Cache disk size
8 vCPUs (2 sockets * 4 cores @ 2.5 GHz)	16 GB	300 GB

The configuration server network settings will be:

Component	Requirement
IP address type	Static
Internet access	The server needs access to these URLs (directly or via proxy): - *.accesscontrol.windows.net - *.backup.windowsazure.com - *.store.core.windows.net - *.blob.core.windows.net - *.hypervrecoverymanager.windowsazure.com - https://management.azure.com - *.services.visualstudio.com - time.nist.gov - time.windows.com OVF also needs access to the following URLs: - https://login.microsoftonline.com - https://secure.aadcdn.microsoftonline-p.com - https://login.live.com - https://auth.gfx.ms - https://graph.windows.net - https://login.windows.net - https://www.live.com - https://www.microsoft.com - https://dev.mysql.com/get/Downloads/MySQLInstaller/mysql-installer-community-5.7.20.0.msi
Ports	443 (Control channel orchestration) 9443 (Data transport)
NIC type	VMXNET3 (if the Configuration Server is a VMware VM)

Key Design Decision 6-13: On-premise Configuration Server

6.4 Azure SQL Database

Azure SQL Database is a relational database-as-a-service (DBaaS) hosted in the Azure cloud. SQL database is built on standardised hardware and software that is owned, hosted, and maintained by Microsoft. Microsoft handles all patching and updating of the SQL database, thereby taking away its management overhead away from Ambulance Victoria.

SQL Database enables organisations to manage unpredictable patterns through "Elastic pools", which dynamically scale up and down the resources required to meet the database demand. This service is offered across three tiers to support lightweight to heavyweight database workloads: Basic, Standard, and Premium. An organisation can allocate performance resources to a pool rather than an individual database, and the pooled databases consume the performance resources of the elastic pool as needed. The charges are applied to the collective performance resources of the pool rather than to a single database performance.

SQL Database provides built-in business continuity and global scalability features, including:

- Automatic backups: SQL Database automatically performs full, differential, and transaction log backups.
- Point-in-time restores: SQL Database supports recovery to any point in time within the automatic backup retention period.
- Active geo-replication: SQL Database allows you to configure up to four readable secondary databases in either the same or globally distributed Azure data centers.
- Failover groups: SQL Database allows you to enable high availability and load balancing at global scale, including transparent geo-replication and failover of large sets of databases and elastic pools.

The following security features are offered by Azure SQL database:

Auditing for compliance and security:

SQL Database Auditing tracks database events and writes them to an audit log in your Azure storage account. Auditing can help you maintain regulatory compliance, understand database activity, and gain insight into discrepancies and anomalies that could indicate business concerns or suspected security violations.

Data encryption at rest:

SQL Database transparent data encryption (TDE) is an encryption-at-rest technology that protects against media theft. It helps protect against the threat of malicious activity by performing real-time encryption and decryption of the database, associated backups, and transaction log files at rest without requiring changes to the application. Customers can manage the TDE encryption keys and other secrets in a secure and compliant manner using Azure Key Vault.

Data encryption in motion

SQL Database is the only database system to offer protection of sensitive data in flight, at rest and during query processing with 'Always Encrypted'.

Dynamic data masking

SQL Database dynamic data masking limits sensitive data exposure by masking it to non-privileged users. It is a policy-based security feature that hides the sensitive data in the result set of a query over designated database fields, while the data in the database is not changed.

Azure Active Directory integration

SQL Database enables you to centrally manage identities of database user and other Microsoft services with Azure Active Directory integration.

Key Design Decision

Azure SQL databases will be utilized for any development in Azure. For IaaS application tiers they will be attached to the data subnet via a Virtual Network Service EndPoint[26] to allow developers access to the databases. For Web Apps they will be deployed within the Web App resource group.

Key Design Decision 6-14: Azure SQL Database

6.4.1 Azure SQL Replication

Active geo-replication[27] is Azure SQL Database feature that allows you to create readable replicas of your database in the same or different data centre (region).

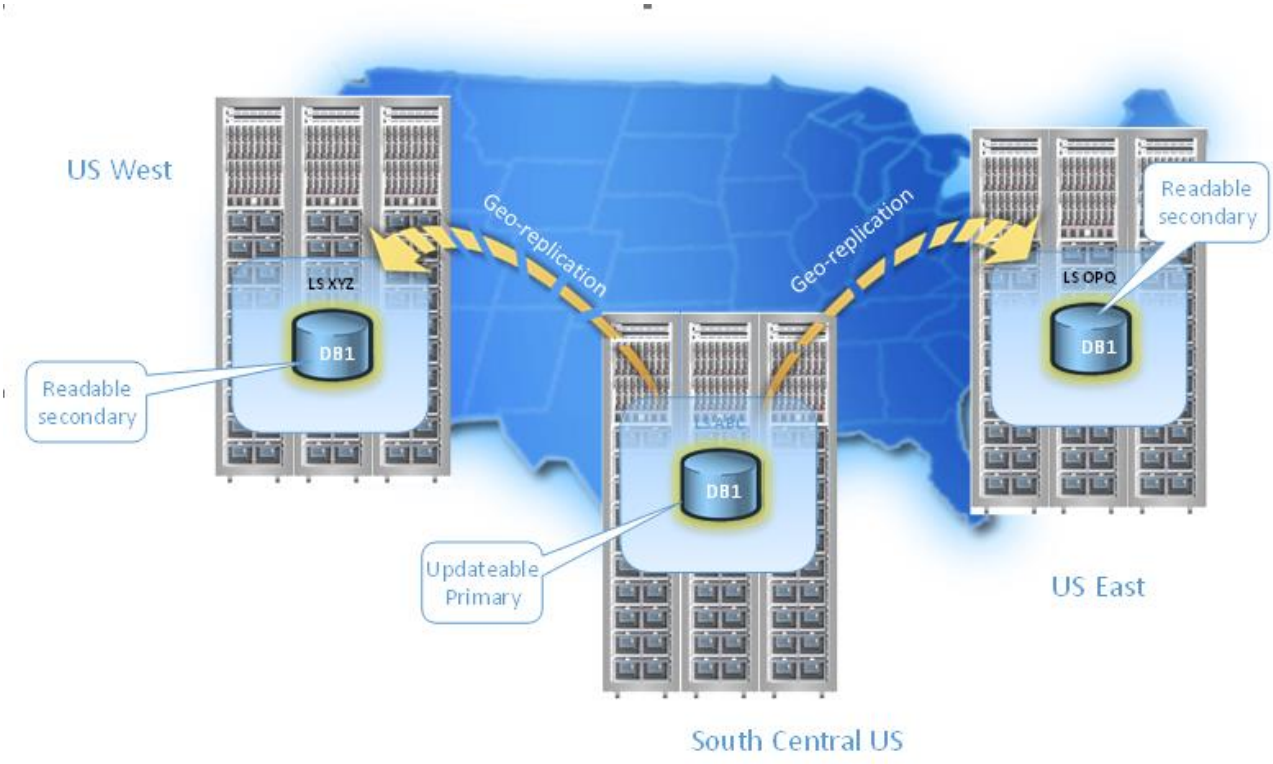

Figure 16 - Azure SQL Replication Model

Active geo-replication is designed as a business continuity solution that allows the application to perform quick disaster recovery in case of a data centre scale outage. If geo-replication is enabled, the application can initiate failover to a secondary database in a different Azure region. Up to four secondaries are supported in the same or different regions, and the secondaries can also be used for read-only access queries. The failover must be initiated manually by the application or the user. After failover, the new primary has a different connection end point.

Key Design Decision
Azure SQL databases will be configured to utilize active geo-replication.

Key Design Decision 6-15: Azure SQL Replication

6.5 Azure Automation

[26] https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-service-endpoints-overview
[27] https://docs.microsoft.com/en-us/azure/sql-database/sql-database-geo-replication-overview

Azure Automation delivers a cloud-based automation and configuration service that provides consistent management across your Azure and non-Azure environments. It consists of process automation, update management, and configuration features.

Process automation

Azure Automation provides you the ability to automate frequent, time-consuming, and error-prone cloud management tasks. This automation helps you focus on work that adds business value. By reducing errors and boosting efficiency, it also helps to lower your operational costs.

Update Management

Update Windows and Linux systems across hybrid environments with Azure Automation. You get visibility of update compliance across Azure, on-premises, and other clouds. You can create schedule deployments to orchestrate the installation of updates within a defined maintenance window.

Configuration Management

Azure Automation desired state configuration is a cloud-based solution for PowerShell DSC that provides services required for enterprise environments. Manage your DSC resources in Azure Automation and apply configurations to virtual or physical machines from a DSC Pull Server in the Azure cloud. It provides rich reports that inform you of major events such as when nodes have deviated from their assigned configuration. You can monitor and automatically update machine configuration across physical and virtual machines, Windows or Linux, in the cloud or on-premises.[28]

Key Design Decision
Two Azure Automation accounts will be deployed to manage selected automation tasks for the Ambulance Victoria Azure environment. Automation accounts are limited in functionality by subscription boundaries so one will be required for the Shared/Production subscription and one for Non-Production. Runbooks will be configured in each subscription to power cycle Azure VMs according to Ambulance Victoria requirements, automating tagging at the component level and to store Network Security Group flow logs. Further runbooks may be required future state to automate tasks for other Azure resources.

Key Design Decision 6-16: Azure Automation

[28] https://docs.microsoft.com/en-us/azure/automation/automation-intro

7 Security Components

7.1 Azure Security Center

Azure Security Center provides unified security management and advanced threat protection across hybrid cloud workloads. With Security Center, you can apply security policies across your workloads, limit your exposure to threats, and detect and respond to attacks.

Security Center features include:

- Centralised policy management – Ensure compliance with company or regulatory security requirements by centrally managing security policies across all your hybrid cloud workloads.
- Continuous security assessment – Monitor the security of machines, networks, storage and data services, and applications to discover potential security issues.
- Actionable recommendations – Remediate security vulnerabilities before they can be exploited by attackers with prioritized and actionable security recommendations.
- Advanced cloud defenses – Reduce threats with just in time access to management ports and whitelisting to control applications running on your VMs.
- Prioritized alerts and incidents - Focus on the most critical threats first with prioritized security alerts and incidents.
- Integrated security solutions - Collect, search, and analyse security data from a variety of sources, including connected partner solutions.

The following security policies can be configured on Azure Security Center

Policy	Description
System updates	Retrieves a daily list of available security and critical updates from Windows Update or Windows Server Update Services and recommends that missing updates be applied.
Security configurations	Analyses operating system configurations daily to determine issues that could make the virtual machine vulnerable to attack.
Endpoint protection	Recommends that endpoint protection be set up for all Windows virtual machines (VMs) to help identify and remove viruses, spyware, and other malicious software.
Disk encryption	Recommends enabling disk encryption in all virtual machines to enhance data protection at rest.
Network security groups	Recommends that network security groups be configured to control inbound and outbound traffic to VMs that have public endpoints.
Web application firewall	Recommends that a web application firewall be set up on virtual machines when either of the following is true: • An instance-level public IP is used, and the inbound security rules for the associated network security group are configured to allow access to port 80/443. • A load-balanced IP is used, and the associated load balancing and inbound network address translation (NAT) rules are configured to allow access to port 80/443. For more information, see Azure Resource Manager support for Load Balancer.

Next generation firewall	Extends network protections beyond network security groups, which are built into Azure. Security Center discovers deployments for which a next generation firewall is recommended, and then you can set up a virtual appliance.
SQL auditing and threat detection	Recommends that auditing of access to your SQL database be enabled for both compliance and advanced threat detection, for investigation purposes.
SQL encryption	Recommends that encryption at rest be enabled for your SQL database, associated backups, and transaction log files. Even if your data is breached, it is not readable.
Vulnerability assessment	Recommends that you install a vulnerability assessment solution on your VM.
Storage encryption	Currently, this feature is available for blobs and Azure Files. After you enable Storage Service Encryption, only new data is encrypted, and any existing files in this storage account remain unencrypted.

The Security Center can be deployed as a free component or a paid premium service. The premium offering current offers no features required by Ambulance Victoria but this may change in the future.

Key Design Decision
The Security Center will be deployed as it is a free service and can be used to easily track whether encryption policies for storage accounts, SQL databases and virtual machines are being followed. It also allows quick visibility of network security elements.

Key Design Decision 7-1: Azure Security Center

7.2 Role Based Access Control (RBAC)

Security-oriented companies should focus on giving employees the exact permissions they need. Too many permissions can expose an account to attackers. Too few permissions mean that employees can't get their work done efficiently. Azure Role-Based Access Control (RBAC) helps address this problem by offering fine-grained access management for Azure.

Using RBAC, you can segregate duties within your team and grant only the amount of access to users that they need to perform their jobs. Instead of giving everybody unrestricted permissions in your Azure subscription or resources, you can allow only certain actions. For example, use RBAC to let one employee manage virtual machines in a subscription, while another can manage SQL databases within the same subscription.

Access Management

Each Azure subscription is associated with one Azure Active Directory (AD) directory. Users, groups, and applications from that directory can manage resources in the Azure subscription. Assign these access rights using the Azure portal, Azure command-line tools, and Azure Management APIs.

Grant access by assigning the appropriate RBAC role to users, groups, and applications at a certain scope. The scope of a role assignment can be a subscription, a resource group, or a single resource. A role assigned at a parent scope also grants access to the children contained within it. For example, a user with access to a resource group can manage all the resources it contains, like websites, virtual machines, and subnets.

The RBAC role that you assign dictates what resources the user, group, or application can manage within that scope.

Built-in Roles

Azure RBAC has three basic roles which apply to all resource types:

- **Owner** has full access to all resources including the right to delegate access to others.
- **Contributor** can create and manage all types of Azure resources but can't grant access to others.
- **Reader** can view existing Azure resources.

The rest of the RBAC roles in Azure allow management of specific Azure resources. For example, the Virtual Machine Contributor role allows the user to create and manage virtual machines. It does not give them access to the virtual network or the subnet that the virtual machine connects to. It is also possible to build out custom roles and apply them to users or application service accounts.

It should be noted that in addition to the roles required for setting up and running an Azure subscription with a multi-tiered environment, application specific roles need to be defined too.

Role	Context	Limitations
Global Administrator (Classic)	Carried over from Office 365 where a Global Administrator may manage any service offered with Office 365. In Azure, can only manage Azure Active Directory (AAD). There may be multiple Global Administrators in a single AAD.	Role may be assigned to an Azure Active Directory user (Service Admin or Co-Admin). Can only be assigned to a User object.
Account Administrator (Classic)	Manages Subscriptions, Service Administrator and Billing Configuration. Exists outside of Azure Subscription. Equivalent of: • **Azure Active Directory Global Administrator (can manage objects in Azure AD);** • **Default Azure Subscription Service Administrator.**	Only 1 per Azure Account. Cannot access Azure Management Portals or APIs (unless left as Service Administrator). Can only be assigned to a User object.
Service Administrator (Classic)	Manages services within an Azure Subscription. Azure Classic Portal and API. Can fully manage (Create/Read/Update/Delete) any service within the Subscription.	Only 1 per Azure Subscription. Cannot manage the parent Account. Can only be assigned to a User object. Assigned global 'Owner' RBAC role by default.
Service Co-Administrators (Classic)	Can manage any service within the Subscription except any Azure Active Directory instance for which the user is not a Global Administrator Azure Classic Portal and API.	Up to 200 users may fill this role per Subscription. Cannot change Service or Account Administrator. Assigned global 'Owner' RBAC role by default.
Owner (RBAC)	Azure RBAC Portal and API Is the equivalent of a Service Admin or Co-Admin in the Classic Portal and API.	If user is added directly as an RBAC-based Owner and is not listed as a Service Admin or Co-Admin, then they may only use the RBAC Portal and API. Can be assigned to User or Group objects.

User Access Administrator (RBAC)	Azure RBAC Portal and API Equivalent of Global Administrator in Classic mode.	Cannot manage Azure Services – only suitable for user and RBAC administration.
Contributor (RBAC)	Azure RBAC Portal and API Equivalent of a Service Co-Admin that does not hold Global Admin rights (i.e. cannot modify access to Azure Service Management).	Cannot manage access to Azure Service Management (Portal or API) – that is, cannot invite other users. Can be assigned to User or Group objects.
Reader (RBAC)	Azure RBAC Portal and API	May only view items in new RBAC Portal and API. Can be assigned to User or Group objects.

Key Design Decision
RBAC will be applied at the subscription level only for super user administrators within the Ambulance Victoria IT team. All other access to Azure resources will be provided at the Resource Group level with permissions provided only to those users who are required to manage those elements via the Azure portal. Custom RBAC roles may be required future state to grant access to specific resources.

Key Design Decision 7-2: Role Based Access Control (RBAC)

8 Identity and Access Management

8.1 Overview

Identity and access management (IAM) is a framework for business processes that facilitates the management of electronic or digital identities. The framework includes the organisational policies for managing digital identity as well as the technologies needed to support identity management.

Identity access management systems should include all the necessary controls and tools to capture and record user login information, manage the enterprise database of user identities and manage the assignment and removal of access privileges. That means that systems used for IAM should provide a centralised directory service with oversight as well as visibility into all aspects of the company user base.

For most enterprise customers IAM is handled by Microsoft's Active Directory product often in conjunction with additional Microsoft or 3rd party products to provide additional federated identity or trusted identity functionality. This can become important with the implementation of software as a service product or implementing organisation wide single sign on[29].

8.2 Active Directory

Active Directory (AD) plays many roles, from being the backbone of distributed security to providing a service-publishing framework. AD provides a central service for administrators to organise network resources, manage users, computers, and applications, as well as to secure intranet and Internet network access.

The virtual machines hosted within Azure should be domain joined where possible. Whilst it is possible to utilise the existing on-premise domain controllers for this purpose requests can be processed faster if Active Directory is deployed within Azure. The extra domain controllers also provide a level of redundancy incase either of the existing datacenters or their associated network links fail.

Key Design Decision
The production Active Directory domains will be extended into the Azure Datacenter to complement the existing infrastructure and to provide authentication and DNS for Azure deployed resources. Both production and non-production virtual machines will utilise the production Active Directory. To prepare the existing Active Directory for extension into the cloud the schema will need to update to allow for Windows 2016 Servers to be deployed and it is recommended the Forest and Domain functional levels be raised to take advantage of new features. For the production domain two servers will be deployed within an Azure availability set to provide high availability. All servers will be configured as additional/backup domain controllers with the FSMO roles remaining within the on-premises co-location facility. Servers will have the following specifications: • D2v3 Instance Type (2vCores, 8GB RAM) • Use 1-year reserved instances • Windows Server 2016 Datacentre • Additional Managed Disk to house sysvol share • Additional disk set to no-caching mode to avoid data loss. • Fixed IP addresses

[29] http://searchsecurity.techtarget.com/definition/identity-access-management-IAM-system

- Deployed into the 'Core Services' subnets in both Australia Southeast and Australia East 'Shared' VNets to provide regional redundancy.
- OMS agent will be deployed to the domain controllers

Key Design Decision 8-1: Active Directory

8.2.1 AD Sites & Services

AD Sites and Services is primarily to locate domain controllers and handle replication priorities. The Azure environment will contain AD domain controllers and will require replication links to be established and integrated into the existin AD Sites & Services topology.

Key Design Decision

The production Active Directory sites and services will be updated to allow a meshed AD replication model by creating replication links between "Azure-to-State DC" and "Azure-to-Eastern Hill DC" as per below:

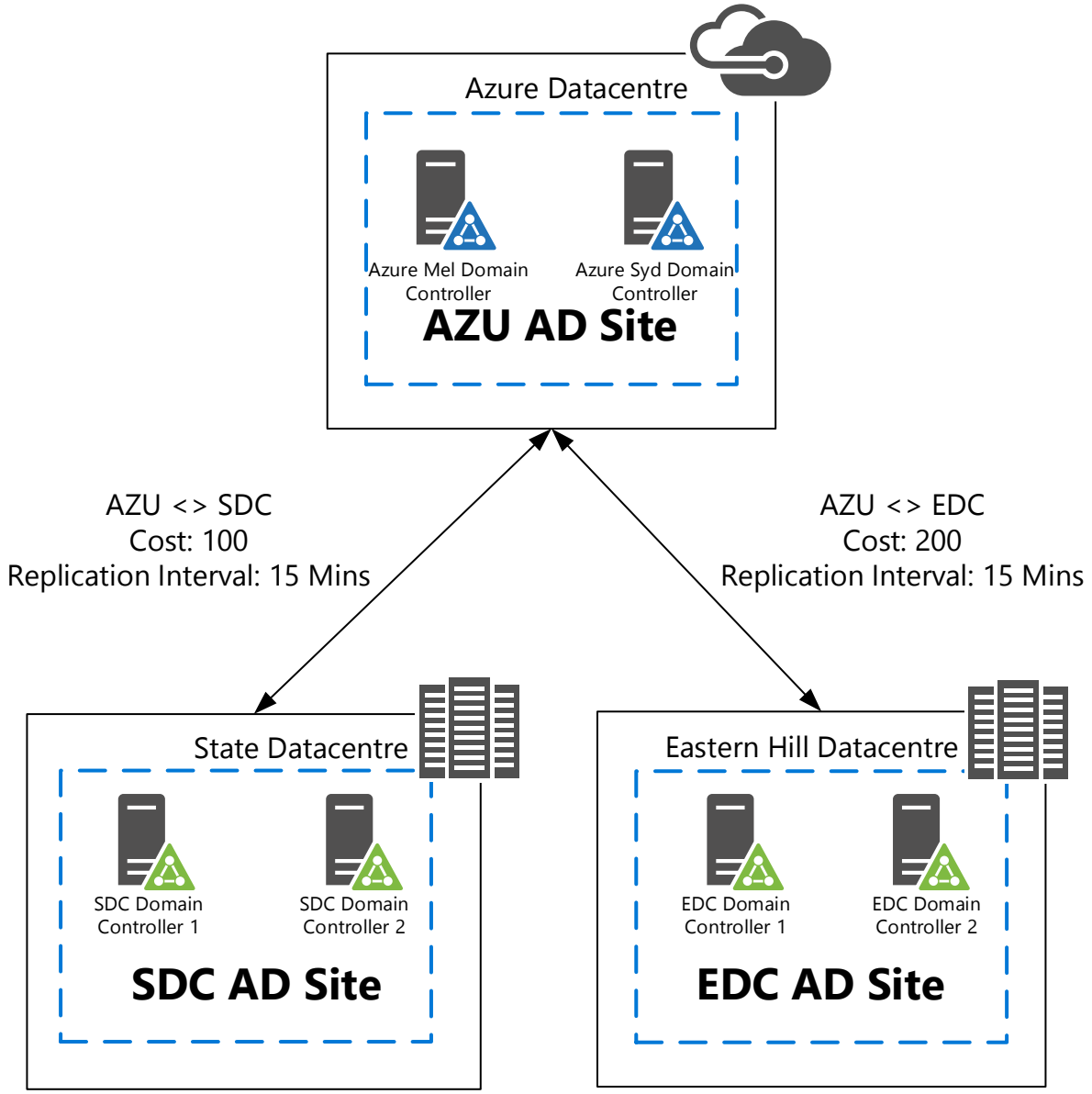

Figure 17 - Active Directory Sites and Services

Key Design Decision 8-2: AD Sites & Services

8.3 Azure Active Directory

Azure Active Directory (Azure AD) is Microsoft's multi-tenant, cloud-based directory and identity management service. Azure AD combines core directory services, advanced identity governance, and application access management. Azure AD also offers a rich, standards-based platform that enables developers to deliver access control to their applications, based on centralized policy and rules[30].

Ambulance Victoria already have Azure Active Directory deployed and synchronised with the corporate Active Directory.

Key Design Decision
The overall solution will utilize the existing Azure Active Directory for integration into VSTS as well as the production and non-production Azure subscriptions. The sandbox subscription will not utilize the corporate Azure AD but will instead utilize the onmicrosoft.com directory created for that subscription as a standalone directory.

Key Design Decision 8-3: Azure Active Directory

8.4 AADC

Azure AD Connect will integrate on-premises Active Directories with Azure Active Directory. This allows you to provide a common identity for users for Office 365, Azure, and SaaS applications integrated with Azure AD.

Users and organizations can take advantage of the following:

- Users can use a single identity to access on-premises applications and cloud services such as Office 365.
- Single tool to provide an easy deployment experience for synchronization and sign-in.
- Provides the newest capabilities for your scenarios. Azure AD Connect replaces older versions of identity integration tools such as DirSync and Azure AD Sync.

Key Design Decision
AADC synchronization is already deployed to provide Active Directory to Azure AD integration for the production domain.

Key Design Decision 8-4: AADC

8.5 ADFS

Identity federation to Azure is provided by the on-premise Active Directory Federation Services (ADFS) infrastructure. This is managed by the Ambulance Victoria infrastructure team. Currently sign-on to Azure leverages this service.

Key Design Decision
ADFS is already deployed to provide single sign-on federation to Azure AD with the corporate Active Directory accounts. The solution will continue to utilize the existing ADFS infrastructure.

Key Design Decision 8-5: ADFS

8.6 Azure Enterprise Applications

Enterprise developers and software-as-a-service (SaaS) providers can develop commercial cloud services or line-of-business applications, that can be integrated with Azure Active Directory (Azure AD) to provide secure sign-in and

[30] https://docs.microsoft.com/en-us/azure/active-directory/active-directory-whatis

authorization for their services. To integrate an application or service with Azure AD, a developer must first register the application with Azure AD.

Recommendation
Ambulance Victoria should utilize the Azure Enterprise Applications[31] technology within Azure to perform future federation to third-parties, SaaS partners and internally developed applications. Ambulance Victoria are licensed with 6900 x Azure AD Basic License which allows 10 apps per user (free tier + Application proxy apps). To fully leverage the Azure AD Enterprise Applications federation capability an upgrade to Azure AD P1 license will enable unlimited integrations (free, Basic tiers + Self-Service App Integration templates)

Key Design Decision 8-6: Azure Enterprise Applications

8.7 MFA

Azure Multi-Factor Authentication (MFA) is Microsoft's two-step verification solution. Azure MFA helps safeguard access to data and applications while meeting user demand for a simple sign-in process. It delivers strong authentication via a range of verification methods, including phone call, text message, or mobile app verification.

Multi-factor authentication (MFA) is currently enabled in the Ambulance Victoria Azure AD using stand-alone licensing. There is an inflight project assessing and addressing MFA.

Key Design Decision
The existing MFA will be utilized for administrative access to the Azure services. When the new MFA solution is deployed this architecture should be reviewed to leverage any future MFA capability.

Key Design Decision 8-7: MFA

[31] https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-integrating-applications

9 Application Hosting Services

9.1 Azure Virtual Machines

An Azure virtual machine (VM) gives you the flexibility of virtualization without having to buy and maintain the physical hardware that runs it. However, you still need to maintain the VM by performing tasks, such as configuring, patching, and installing the software that runs on it.

Azure virtual machines can be used in several ways. Some examples are:

- Development and test – Azure VMs offer a quick and easy-way to create a computer with specific configurations required to code and test an application.
- Applications in the cloud – Because demand for your application can fluctuate, it might make economic sense to run it on a VM in Azure. You pay for extra VMs when you need them and shut them down when you don't.
- Extended datacenter – Virtual machines in an Azure virtual network can easily be connected to your organization's network.

The number of VMs that your application uses can scale up and out to whatever is required to meet your needs.[32]

9.1.1 Virtual Machine Specifications

Ambulance Victoria have decided on three types of virtual machine configurations: Small, Medium and Large. These requirements were compared against the Azure compute offerings that closely aligned to their CPU and memory needs:

Type	Azure Model	CPU	Memory (GB)	Disk (GB)	NIC / Bandwidth
Small	Standard_A2_v2	2	4	20	2 / moderate
Medium	Standard_A4_v2	4	8	40	4 / high
Large	Standard_A8_v2	8	16	80	8 / high

Figure 18 – Virtual Machine Specification Mapping

Key Design Decision
The following Azure models will be configured for selection in the ARM templates: • Standard_A2_v2 • Standard_A4_v2 • Standard_A8_v2

Key Design Decision 9-1: Virtual Machine Specifications

9.1.2 Virtual Machine Operating Systems

Key Design Decision
By preference all Azure virtual machines will be deployed with a Windows Server 2016 Datacenter Edition based image that is available via the Azure marketplace. If the application requires an older version, then Windows Server 2012 R2 based servers can be deployed.

Key Design Decision 9-2: Virtual Machine Operating Systems

[32] https://docs.microsoft.com/en-us/azure/virtual-machines/windows/overview

9.1.2.1 Windows Updates

Windows Server Update Services (WSUS)[33] enables information technology administrators to deploy the latest Microsoft product updates. You can use WSUS to fully manage the distribution of updates that are released through Microsoft Update to computers on your network. This topic provides an overview of this server role and more information about how to deploy and maintain WSUS

Ambulance Victoria currently utilise the Windows Update Service to deploy updates across all virtual machines.

Key Design Decision
Windows Server Update Service will be utilised to perform the Windows Server updates on the virtual machines.

Key Design Decision 9-3: Windows Updates

Recommendation
It is recommended to transition to Azure Automation Update Management service to simplify administration and also get in-depth compliance reporting and integrate into the Azure Security Centre.

Recommendation 9-1: Windows Updates

9.1.3 Azure Virtual Machine Extensions for Windows

Azure virtual machine (VM) extensions are small applications that provide post-deployment configuration and automation tasks on Azure VMs, you can use existing images and then customize them as part of your deployments, getting you out of the business of custom image building.

Key Design Decision
Virtual Machine Extensions will be used to deploy Standard Operating Environment (SOE) tools and applications to each Windows virtual machine wherever possible. E.g. the SCCM agent will be deployed via an extension then the SCCM agent will deploy the Windows Defender product.

Key Design Decision 9-4: Azure Virtual Machine Extensions

9.1.4 Azure Virtual Machine Configuration for Linux

Ansible[34] is open source software that automates software provisioning, configuration management, and application deployment. Ansible connects via SSH, remote PowerShell or via other remote APIs.

Ambulance Victoria currently utilise the Ansible to deploy updates and configuration across all Linux virtual machines.

Key Design Decision
Ansible will be used to deploy Standard Operating Environment (SOE) tools and applications to each Linux virtual machine wherever possible.

Key Design Decision 9-5 - Azure Virtual Machine Configuration for Linux

9.1.5 Virtual Machine Antimalware Protection

Anti-malware is software tools and programs designed to identify and prevent malicious software, or malware, from infecting computer systems or electronic devices.

[33] https://docs.microsoft.com/en-us/windows-server/administration/windows-server-update-services/get-started/windows-server-update-services-wsus

[34] https://www.ansible.com/

Ambulance Victoria currently utilise the Windows Defender product deployed via SCCM and they require protection across all Windows server virtual machines. Linux virtual machines do not run antimalware products.

Windows Defender Antivirus for Windows Server

Windows Server 2016 now includes Windows Defender Antivirus[35]. Windows Defender AV is malware protection that immediately and actively protects Windows Server 2016 against known malware and can regularly update antimalware definitions through Windows Update.

Microsoft Antimalware for Azure

Microsoft Antimalware for Azure[36] is a free real-time protection that helps identify and remove viruses, spyware, and other malicious software. It generates alerts when known malicious or unwanted software tries to install itself or run on your Azure systems.

The solution is built on the same antimalware platform as Microsoft Security Essentials [MSE], Microsoft Forefront Endpoint Protection, Microsoft System Center Endpoint Protection, Windows Intune, and Windows Defender. Microsoft Antimalware for Azure is a single-agent solution for applications and tenant environments, designed to run in the background without human intervention. Protection may be deployed based on the needs of application workloads, with either basic secure-by-default or advanced custom configuration, including antimalware monitoring.

Key Design Decision
Windows Defender Antivirus will be deployed to the Windows virtual machines and updated using the existing SCCM process No antimalware will be deployed to the Linux virtual machines

Key Design Decision 9-6: Virtual Machine Antimalware Protection

Recommendation
To simplify the integration requirements as there are no firewall ports required to be opened, it is recommended that Microsoft Antimalware for Azure should be deployed to all new Windows virtual machines as an Azure Virtual Machine Extension[37]. This will also simplify administration and provide in-depth compliance reporting with integration into the Azure Security Centre.

Recommendation 9-2 - Virtual Machine Antimalware Protection

9.2 Azure App Service

Azure App Service Web Apps[38] (or just Web Apps) is a service for hosting web applications, REST APIs, and mobile back ends. You can develop many different languages, be it .NET, .NET Core, Java, Ruby, Node.js, PHP, or Python. Applications run and scale with ease on Windows-based environments.

With App Service, you pay for the Azure compute resources you use. The compute resources you use is determined by the App Service plan that you run your Web Apps on. For more information, see App Service plans in Azure Web Apps.

Key Design Decision

[35] https://docs.microsoft.com/en-us/windows-server/security/windows-defender/windows-defender-overview-windows-server
[36] https://docs.microsoft.com/en-us/azure/security/azure-security-antimalware
[37] https://docs.microsoft.com/en-us/azure/virtual-machines/extensions/iaas-antimalware-windows
[38] https://docs.microsoft.com/en-us/azure/app-service/app-service-web-overview

The new Azure platform and subscription will utilise Azure App Service Web Apps as a preferred technology for when coding services are procured. Each WebApp will be deployed into the resource group for the "service" as defined by the IT business service catalogue.

Recommendation 9-3: Azure App Service

9.2.1 Azure Application Service Environments

The Azure App Service Environment is an Azure App Service feature that provides a fully isolated and dedicated environment for securely running App Service apps at high scale. This capability can host your:

- Windows web apps
- Linux web apps (in Preview)
- Docker containers (in Preview)
- Mobile apps
- Functions

App Service environments (ASEs) are appropriate for application workloads that require:

- Very high scale.
- Isolation and secure network access.
- High memory utilization.

Customers can create multiple ASEs within a single Azure region or across multiple Azure regions. This flexibility makes ASEs ideal for horizontally scaling stateless application tiers in support of high RPS workloads.

ASEs are isolated to running only a single customer's applications and are always deployed into a virtual network. Customers have fine-grained control over inbound and outbound application network traffic. Applications can establish high-speed secure connections over VPNs to on-premises corporate resources.

Key Design Decision
There is no current use case for high throughput or a large user base that warrants a very scalable web farm, nor is there a desire to use legacy integration techniques (i.e. internal network communication vs modern API based communication) for web application development. So, App Service environments (ASEs) were not deemed appropriate as ASE's are for application workloads that require very high scale, isolation and secure network access, and High memory utilization.

Key Design Decision 9-7: Azure Application Service Environments

9.3 Azure Functions

Azure Functions[39] is a solution for easily running small pieces of code, or "functions," in the cloud. You can write just the code you need for the problem at hand, without worrying about a whole application or the infrastructure to run it. Functions can make development even more productive, and you can use your development language of choice, such as C#, F#, Node.js, Java, or PHP. Pay only for the time your code runs and trust Azure to scale as needed. Azure Functions lets you develop serverless applications on Microsoft Azure.

Azure Functions has two kinds of pricing plans:

[39] https://docs.microsoft.com/en-us/azure/azure-functions/functions-overview

Consumption plan - When your function runs, Azure provides all of the necessary computational resources. You don't have to worry about resource management, and you only pay for the time that your code runs.

App Service plan - Run your functions just like your web, mobile, and API apps. When you are already using App Service for your other applications, you can run your functions on the same plan at no additional cost.

Key Design Decision
Ambulance Victoria should utilise Azure Functions for small code deployments and/or as a capability offload from the main core application. The pricing plan should be determined at the function design phase by what suits the technical requirements of the function and its relationship to any existing App Service Plans. E.g. if there is an existing App Service Plan and the function that is being developed will form part of an overall service it should be included in that App Service Plan rather than use a consumption plan. Requests to Azure Functions have built in security so can be accessed directly without traversing via the WAF.

Key Design Decision 9-8: Azure Functions

9.4 API Management

API Management (APIM) helps organizations publish APIs to external, partner, and internal developers to unlock the potential of their data and services.

Ambulance Victoria have an existing Azure API Management service running in the Azure Service Manager tenancy. This is linked to the Enterprise Service Bus and is utilised for when API's need to be public facing. This service is currently running with a developer license.

Key Design Decision
Internal APIs will be managed by Ambulance Victoria's ESB platform. External APIs i.e. Public IaaS or Public PaaS, will be created on the existing API Management by the application development team.

Key Design Decision 9-9: API Management

Recommendation
It is recommended that a review of the current Azure API Management service is performed and rearchitected to apply full production rigor to the service.

Recommendation 9-4: API Management

9.5 Azure Service Fabric

Ambulance Victoria have expressed interest in containerisation technology and microservices architecture. Azure Service Fabric[40] is a distributed systems platform that makes it easy to package, deploy, and manage scalable and reliable microservices and containers. Service Fabric also addresses the significant challenges in developing and managing cloud native applications. Developers and administrators can avoid complex infrastructure problems and focus on implementing mission-critical, demanding workloads that are scalable, reliable, and manageable.

Container deployment and orchestration

[40] https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-overview

Service Fabric is Microsoft's container orchestrator deploying microservices across a cluster of machines. Microservices can be developed in many ways from using the Service Fabric programming models, ASP.NET Core, to deploying any code of your choice. Importantly, you can mix both services in processes and services in containers in the same application. If you just want to deploy and manage containers, Service Fabric is a perfect choice as a container orchestrator.

Any OS, any cloud

Service Fabric runs everywhere. You can create clusters for Service Fabric in many environments, including Azure or on premises, on Windows Server, or on Linux. You can even create clusters on other public clouds. In addition, the development environment in the SDK is identical to the production environment, with no emulators involved. In other words, what runs on your local development cluster deploys to the clusters in other environments.

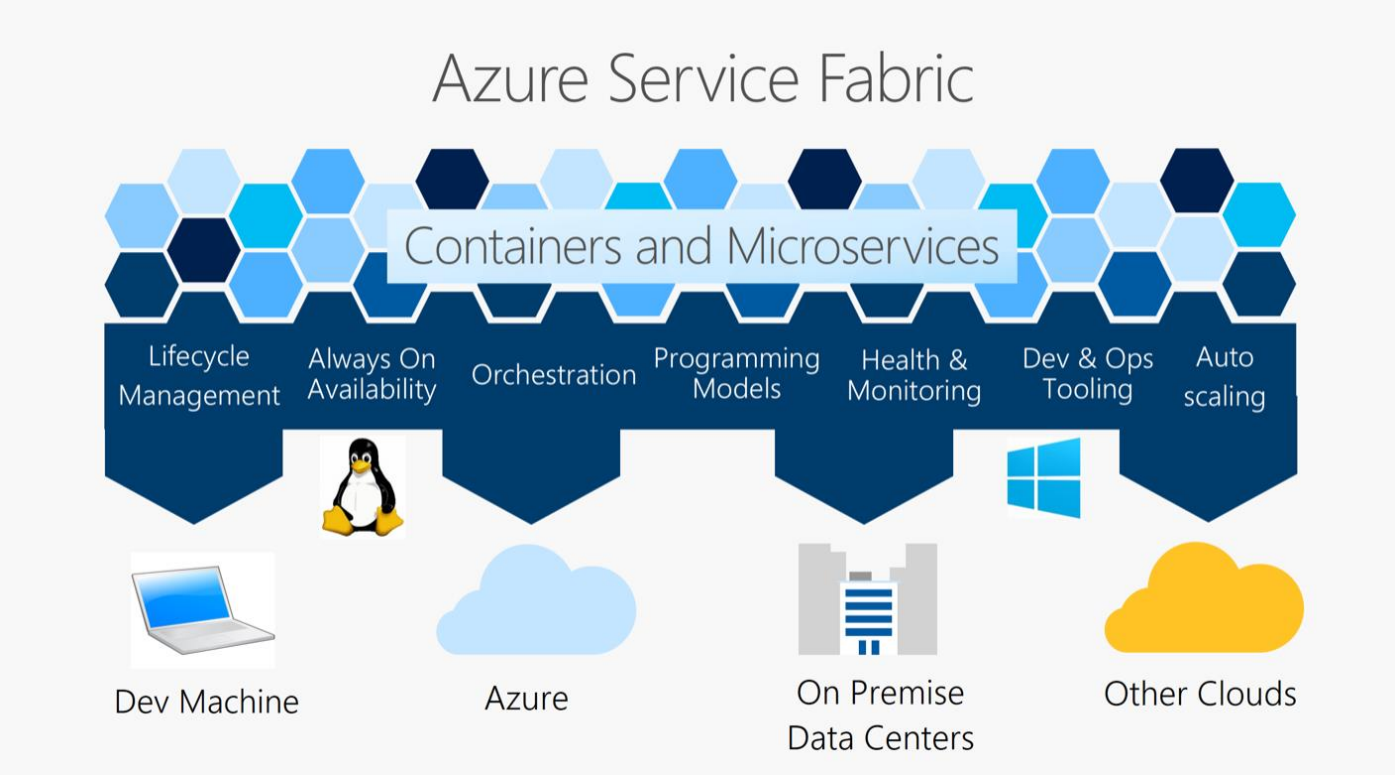

Figure 19 - Azure Service Fabric Infographic

Recommendation
Ambulance Victoria should pilot the use of Azure Service Fabric and test its "Any platform" interoperability by deploying on a developer's local machine using the SDK and then deploying to Azure cloud. It can then be determined whether it should be integrated into the deployment tool as part of the application development strategy.

Recommendation 9-5: Azure Service Fabric

10 Deployment Tools

10.1 Overview

The delivery mechanism is an end-to-end integrated capability to deliver new applications, modifications to applications and/or infrastructure into the Azure environment. It encompasses different people, processes and technologies to help automate, streamline and govern Ambulance Victoria's release pipeline to ensure services are consistently released to the organisations best-practise.

At a high level it can be summarised as follows:

- A VSTS Git repository for version control of the infrastructure ARM templates and scripts.
- A code editing tool and local Git repository tool for your local machine.
- Visual Studio Team Services (VSTS) to manage the continuous integration / continuous delivery (CICD) pipeline.
- Approval workflows within VSTS and key stage gates for releases to be managed into production by the infrastructure team.
- Using deployment patterns to build a library of reusable code to deploy well governed foundation architectures quickly.
- Automating the deployment pattern through ARM templates, scripts and application code.

The industry trend for cloud-based infrastructure deployments is to deploy Infrastructure as Code (IAC).

Delivering your infrastructure as code lets you adopt agile practices and deliver software more easily, rapidly, safely & reliably. Infrastructure as code is the prerequisite for common DevOps practices such as version control, code review, continuous integration and automated testing.

Key Design Decision
All infrastructure within Azure should be delivered as Infrastructure as Code (IAC) rather than manual configuration.

Key Design Decision 10-1: Deployment Tools

10.2 Infrastructure Coding Language

With Azure Resource Manager[41], you can create a template (in JSON format) that defines the infrastructure and configuration of your Azure solution. By using a template, you can repeatedly deploy your solution throughout its lifecycle and have confidence your resources are deployed in a consistent state. When you create a solution from the portal, the solution automatically includes a deployment template. You don't have to create your template from scratch because you can start with the template for your solution and customize it to meet your specific needs. You can retrieve a template for an existing resource group by either exporting the current state of the resource group, or viewing the template used for a particular deployment. Viewing the exported template is a helpful way to learn about the template syntax.

An additionally coding tool is PowerShell[42] which is built on the .NET Framework, PowerShell is a task-based command-line shell and scripting language; it is designed specifically for system administrators and power-users, to rapidly automate the administration of multiple operating systems (Linux, macOS, Unix, and Windows) and the processes

[41] https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-overview#template-deployment
[42] https://docs.microsoft.com/en-us/powershell/scripting/powershell-scripting

related to the applications that run on those operating systems. PowerShell can be used to run commands against Azure and the various services within it in addition to managing existing on-premise Microsoft services.

Key Design Decision
Azure Resource Manager (ARM) templates in JSON format will be the primary coding language for all infrastructure deployments. Powershell will be used as a standalone or to complement ARM templates to help facilitate tasks that cannot be managed via ARM templates. Linux OS configuration will utilize Ansible.

Key Design Decision 10-2: Infrastructure Coding Language

10.3 Visual Studio Team Services

10.3.1 Overview

Visual Studio Team Services[43] (VSTS) is a cloud-based product that allows you to create a full CI/CD pipeline for your application–no matter which language you're using—and deploy to several targets, including virtual machines, Azure Service Fabric, and Azure App Services.

It is Microsoft's DevOps solution for Azure. The Build and Release services provide streamlined experiences to deploy your apps to one of Azure's many services. These services provide a continuous delivery solution if your code is managed in GitHub, VSTS Git or Team Foundation Version Control, or in one of the other Git servers.

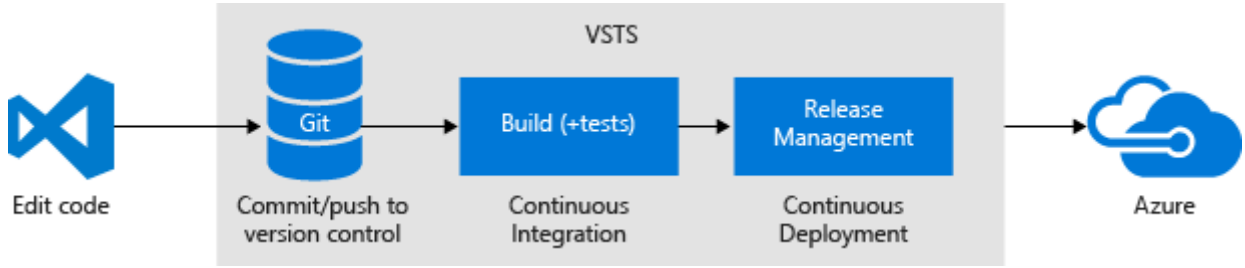

Figure 20 - Release Process Diagram

After you commit and push a code change, it can be automatically built and then deployed. The results can automatically show up on your site.

10.3.2 Code Repository and Source Control (Git)

Repositories and Source Control allow you to manage your code in Git version control with VSTS. You can use version control to save your work and coordinate code changes across your team.

VSTS has a native Git repository that allows you to perform edits directly in VSTS as well as perform pull requests from connected Git sources.

Key Design Decision
A VSTS Git repository will be deployed and configured as the central infrastructure code repository.

Key Design Decision 10-3: Code Repository and Source Control (Git)

[43] https://docs.microsoft.com/en-us/vsts/deploy-azure/?view=vsts

10.3.2.1 Code Commits

Code edits can be performed within the VSTS portal, but it is much easier to utilise personal computers running a code editing tool of your choice to perform edits on your own machine and also a Git client of your choice to commit and push those changes to the central source repository.

10.3.2.1.1 Visual Studio Code

Visual Studio Code[44] is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

10.3.2.1.2 Sourcetree

Sourcetree[45] is a free Git client for Windows and Mac. It simplifies how you interact with your Git repositories so you can focus on coding and visualize and manage your repositories through Sourcetree's simple Git GUI.

Recommendation
It is recommended to deploy to administrator's desktops; a code editing tool (e.g. Microsoft's Visual Code) and a Git client for committing changes to the Git repository (e.g. Atlassian's Sourcetree).

Recommendation 10-1: Code Commits

10.3.2.2 Pull Request Approvals

Code within Git can be committed and merged back[46] into the main code repository within VSTS Git. Branch policies[47] are an important part of the Git workflow and enable you to:

- Isolate work in progress from the completed work in your master branch
- Guarantee changes build before they get to master
- Limit who can contribute to specific branches
- Enforce who can create branches and the naming guidelines for the branches
- Automatically include the right reviewers for every code change
- Enforce best practices with required code reviewers

Key Design Decision
A VSTS Git branch policy to enforce reviews should be configured on all master branches for core infrastructure. Initially this will provide a basic level of protection from catastrophic errors and in future the policy can be adjusted based on operational needs e.g. enforce optimal coding practices. The code reviews will be performed by the cloud infrastructure engineer.

Key Design Decision 10-4: Pull Request Approvals

10.3.3 Build and Release

Visual Studio Team Services (VSTS) is Microsoft's DevOps solution for Azure. The Build and Release services provide streamlined experiences to deploy your apps to one of Azure's many services. These services provide a continuous delivery solution if your code is managed in VSTS Git or in one of the other Git servers.

---

[44] https://code.visualstudio.com/docs
[45] https://www.sourcetreeapp.com/
[46] https://docs.microsoft.com/en-us/vsts/git/tutorial/pullrequest?view=vsts
[47] https://docs.microsoft.com/en-us/vsts/git/branch-policies-overview?view=vsts

10.3.3.1 Automated Builds (Continuous Integration)

Continuous Integration (CI) is the practice used by development teams to automate the merging and testing of code. Implementing CI helps to catch bugs early in the development cycle, which makes them less expensive to fix. Automated tests execute as part of the CI process to ensure quality. Artefacts are produced from CI systems and fed to release pipelines to drive frequent deployments. The Build service in VSTS helps you set up and manage CI for your applications.

Artefacts are the files that you want your build to produce. Artefacts can be nearly anything your team needs to test or deploy your app. For example, you've got a .DLL and .EXE executable files and .PDB symbols file of a C# or C++ .NET Windows app.

Once a code is pulled into the VSTS Git repository it a trigger can be set in the build to automatically create the artefacts ready for consumption by the release process.

Key Design Decision
Separate builds will be created for application code and infrastructure code. These can then be fed into the release management pipeline in VSTS together. Builds will be set to automatically trigger the creation of the artefacts where possible.

Key Design Decision 10-5: Continuous Integration

10.3.3.2 Automated Releases (Continuous Delivery)

Continuous Delivery (CD) is a process by which code is built, tested, and deployed to one or more test and production environments. Deploying and testing in multiple environments drives quality. CI systems produce the deployable artefacts (builds) including infrastructure and apps. Automated release pipelines consume these artefacts to release new versions and fixes to existing systems. The Release service in VSTS helps you set up and manage CD for your applications.

10.3.3.2.1 CD Environments

An environment is a logical and independent entity that represents where you want to deploy a release generated from a release pipeline. Releases should be configured to align to how the organisation releases their applications from development-to-production.

Key Design Decision
Four environments will be created in each new VSTS project: • DEV • TST • UAT • PRD The flow of a release will be as follows: DEV → TST → UAT → PRD

Key Design Decision 10-6: CD Environments

10.3.3.2.2 Release Triggers and Approvals

In your release pipeline you can implement various checks and conditions to control the deployment.

- Set branch filters to configure the continuous deployment trigger on the artefact of the release pipeline.

- Set pre-deployment approvals as a pre-condition for deployment to an environment.
- Configure gates as a pre-condition for deployment to an environment.
- Specify conditions for a task to run.

Additionally, the release can be configured to automatically trigger once a build has completed.

Key Design Decision
The following release design will be utilized to allow developers to authorize their own releases into DEV but still manually trigger the deployment within VSTS i.e. they won't just write a code change and it'll automatically run the release, they will have to kick it off. The release structure will also automatically trigger TST, UAT and PRD but each environment will require a pre-deployment approval from the application and infrastructure leads. PRD will require an authorized change record.

Environment	Trigger	Pre-deployment Approval
DEV	Manual	None
TST	Automatic after DEV success	Infrastructure Leads
UAT	Automatic after TST success	Infrastructure Leads
PRD	Automatic after UAT success	Change Management Approval, Application & Infrastructure Leads

Key Design Decision 10-7: Release Triggers and Approvals

10.3.4 Integration

10.3.4.1 VSTS Identity and Access Management

Security within VSTS is critical to the integrity of the IT systems and Azure environment. One of the main principles for IAC is to allow the final release into production to be managed by the Infrastructure team.

Key Design Decision
The Infrastructure Team will manage permissions within VSTS and will provision new projects on behalf of any requestors with the appropriate rights to projects.

Key Design Decision 10-8: VSTS Identity and Access Management

10.3.4.2 Azure Service Integration

Each VSTS project will require an Azure service connection. The Azure service connection stores the credentials to connect from VSTS to Azure and allow VSTS to deploy the code into Azure.

Service connections should only be created once the release approval model has been established.

Key Design Decision
For each project, two service connections should be created in VSTS. One for the non-production subscription and one for the production subscription. The non-production service connection will be configured in the following VSTS Environments: DEV, TST & UAT The production service connection will be configured in the following VSTS Environments:

PRD
The service connection should use the principle of least privilege so the corresponding account in Azure should only be granted permissions to the Azure resource group that will contain the VSTS project.

Key Design Decision 10-9: Azure Service Integration

Appendix A - Requirements Traceability

The requirements traceability is used to track how many requirements solution solves for. The overall total of requirements met is ~80%.

All Requirements by Status

The following chart is the summary of all the requirements by their status.

Requirement Status	Count
Partially	18
Not met	16
Met	128
Grand Total	162

All Requirements by Criticality

This next table is a summary of the status of requirements from a criticality point of view.

Requirement Criticality	Count
DESIRABLE	87
Partially	10
Not met	6
Met	71
MANDATORY	65
Partially	8
Met	57
OPTIONAL	10
Not met	10
Grand Total	162

Figure 21 - Traceability Table by the Criticality of the Requirement

Requirements Traceability Matrix

Req. ID	Requirement	Met?	Notes
REQ001	The solution MUST provide DR capability for platinum and gold services that are hosted in the cloud	P	Patterns only established for Platinum
REQ009	The solution MUST provide DR capability for platinum and gold services that are hosted on-premise, pending DR patterns from EA	P	Patterns only established for Platinum
REQ002	The solution MUST support established AV patterns for HA and DR	P	Patterns only established for Platinum
REQ003	The solution MUST provide HA capability for platinum and gold services that are hosted in the cloud	P	Patterns only established for Platinum
REQ004	The solution MUST provide a secure environment	Y	
REQ005	The solution MUST support Infrastructure as Code	Y	
REQ006	The solution MUST support the EA principle of cloud first, mobile first	Y	Enables development of

			cloud scalable and internet facing services
REQ007	The solution MUST support the EA principle of cloud is the future	Y	
REQ008	The solution MUST support the EA principle of SaaS first then PaaS then IaaS	Y	
REQ010	The solution SHOULD NOT provide duplication of services	P	
REQ011	The solution SHOULD provide security conscious processes for the management of the environment in additional to technical controls	Y	
REQ012	The solution SHOULD utilise "best of breed" technology preference	Y	
REQ013	The solution SHOULD support AV to be more flexible	Y	
REQ015	The solution SHOULD provide ease of deployment	Y	
REQ016	The solution SHOULD support ease of use philosophy	Y	
REQ017	The solution MAY provide a road map for delivery of the end state solution	N	
REQ021	The solution MUST utilise the existing API management service	Y	
REQ018	The solution SHOULD support API Management capability	N	
REQ019	The solution SHOULD provide API Management configuration best practise	N	
REQ020	The solution SHOULD provide simplified API management capability in regard to network layer integration	N	
REQ022	The solution MAY provide an alternative technology to the Oracle stack	N	
REQ023	The solution MAY provide recommendations for a service bus technology i.e. Service Bus + API Management + Identity + Security + Data Classification + Storage + Analytics + Streaming	N	
REQ024	The solution MUST provide billing visibility for IT	Y	
REQ025	The solution MUST provide cost management capability to help AV perform charge back	Y	
REQ026	The solution MUST provide a tagging taxonomy to apportion costs - down to systems or business	Y	
REQ027	The solution SHOULD provide billing visibility for business owners	Y	
REQ028	The solution SHOULD provide connectivity that leverages the Telstra TPAMS agreement	P	
REQ029	The solution SHOULD provide dev/test licensing model (if cost advantageous)	Y	
REQ030	The solution SHOULD utilise EA pricing benefits	Y	
REQ031	The solution SHOULD support optimisation of costs for permanent infrastructure virtual machines and SQL services	Y	i.e. reserved instances
REQ032	The solution SHOULD provide the ability to demonstrate OOM costing for future proposed services	Y	
REQ033	The solution SHOULD support the ability to forecast EA discounts to OOM costing	Y	
REQ034	The solution SHOULD provide the ability to perform cost management reporting	Y	
REQ035	The solution MAY provide cost analysis assessment for MSDN licensing (licensing is generally out-of-scope)	N	
REQ036	The solution MAY provide cost management for the network hybrid connectivity	N	
REQ037	The solution MUST provide a standard offering of SQL on PaaS and should be enforced	Y	
REQ038	The solution MUST support hosting SQL databases	Y	
REQ039	The solution SHOULD provide an architecture that complies with Ambulance Victoria's data classification policies	Y	

REQ040	The solution SHOULD provide access to the PaaS database services for Azure virtual machines where required	Y	i.e. virtual network endpoints
REQ041	The solution SHOULD NOT provide Azure Cosmos DB as there is no requirement for non-relational databases	Y	
REQ042	The solution SHOULD support hosting MySQL databases	P	Technology is supported in Azure
REQ043	The solution SHOULD provide LRS as standard and GRS as an option	Y	
REQ044	The solution SHOULD NOT provide publicly accessible storage options	Y	
REQ045	The solution SHOULD provide redundant storage where possible and depending on HA/DR requirements	Y	
REQ046	The solution MAY provide technology for a data warehouse capability	N	
REQ047	The solution MUST provide fit-for-purpose Paas/SaaS components (instead of the current development subscription of API management being used for production)	Y	
REQ048	The solution MUST provide patterns for internal facing web services hosted on IaaS	Y	
REQ049	The solution MUST provide patterns for public facing web services hosted on IaaS	Y	
REQ050	The solution MUST provide patterns for public facing web services hosted on PaaS	Y	
REQ051	The solution MUST provide a build and release pipeline for infrastructure as code	Y	
REQ052	The solution MUST provide domain joined servers as a standard IaaS offering	Y	
REQ053	The solution SHOULD provide alignment to the application build and release pipeline	Y	
REQ054	The solution SHOULD support compatibility for integration of new SaaS services	Y	
REQ055	The solution SHOULD provide flexibility to provision services i.e. know it is available to go by ensuring the underpinning infrastructure is ready	Y	
REQ056	The solution SHOULD provide integration into selected existing SaaS services	Y	
REQ057	The solution SHOULD provide patterns for internal facing web services hosted on PaaS (where supported)	N	
REQ058	The solution SHOULD support utilising PaaS in non-function requirements for procurement of application outsourcing tender processes	Y	
REQ059	The solution SHOULD provide benefits and rationalisation for Infrastructure as Code	N	
REQ060	The solution SHOULD provide code repository solution for infrastructure code	Y	
REQ061	The solution SHOULD provide infrastructure as code release controls	Y	
REQ062	The solution SHOULD provide a standard set of offerings for virtual machine types SMALL | MEDIUM | LARGE | CUSTOM	Y	
REQ063	The solution SHOULD provide a standard set of sizes for with minimum of 2-4 vCPUs | 4-8GB RAM | 64GB HDD	Y	
REQ064	The solution SHOULD utilise the existing group policy infrastructure	Y	
REQ065	The solution SHOULD provide Windows Server 2016 as a default operating system platform	Y	
REQ066	The solution MUST provide a disaster recovery solution for Azure virtual machines systems	Y	

REQ067	The solution MUST provide a disaster recovery solution that meets the service catalog RPO and RTO metrics for a Platinum service built in Azure	P	
REQ068	The solution MUST support the principle of data in the cloud should be backed up in cloud	P	
REQ069	The solution SHOULD provide a disaster recovery solution for supported on-premise systems	Y	
REQ070	The solution SHOULD provide a disaster recovery solution that meets the service catalog RPO and RTO metrics for a Platinum service built on-premise	P	
REQ071	The solution SHOULD NOT provide an Azure Recovery Vault as not requirement since CommVault will be utilised	Y	Needs to be architected by Perfect, leverage integration agents
REQ072	The solution SHOULD utilise CommVault agents that are pushed to the servers	Y	
REQ073	The solution SHOULD utilise the overall CommVault solution to provide a single pane of glass view for backups	Y	SQL backups? Can commvault back up?
REQ074	The solution MUST utilise Azure AD as the federation technology of choice	Y	
REQ075	The solution MUST utilise the existing MFA model for administrator accounts	Y	
REQ076	The solution MUST provide Active Directory domain controllers in each region	Y	
REQ077	The solution MUST utilise the existing ADFS service running on the VMs in ASM	Y	
REQ078	The solution MUST utilise the existing administrator account model i.e. separate ADM accounts	Y	
REQ079	The solution MUST utilise the existing service account model	Y	
REQ080	The solution MUST provide an RBAC framework for the Azure EA portal, subscriptions, resource groups and Azure services	Y	roles based access control explicit permissions down to the subscription level and core resource group level. Implied permissions will be granted to the application / service resource groups
REQ081	The solution MUST provide an RBAC model that supports delegating to access to third-parties	Y	
REQ082	The solution MUST provide secure repository controls	Y	
REQ083	The solution MUST provide the systems engineering team as the storage administrators	Y	
REQ084	The solution SHOULD provide AAD Identity, Authentication and Integration	Y	
REQ085	The solution SHOULD utilise the existing Azure AD Connect service	Y	
REQ086	The solution SHOULD support utilising AAD SSO in non-function requirements for procurement of application outsourcing tender processes	Y	
REQ087	The solution SHOULD provide AD sites and services as a combined site between Melb and Syd	Y	

REQ088	The solution SHOULD support SCIM integration technology	P	
REQ089	The solution SHOULD NOT provide a priviledged access management solution as one is on the horizon	Y	
REQ090	The solution SHOULD provide rescue accounts for critical services	P	
REQ091	The solution SHOULD provide well defined roles, responsibilities, security rules, network design	P	
REQ092	The solution MAY provide Business to consumer integration	N	
REQ093	The solution MUST NOT provide a service management ticket integration solution	Y	
REQ094	The solution MUST provide monitoring capabilities	Y	
REQ095	The solution MUST provide OMS integration for the domain controllers to send security logs to the Log Rhythm SIEM	Y	
REQ096	The solution MUST utilise the Log Rhythm security log aggregation tool i.e. ports must be opened	Y	
REQ097	The solution SHOULD NOT provide a customised event management system as this will be performed by the future event management project	Y	
REQ098	The solution SHOULD provide provide security log aggregation into Log Rhythm for all eligible services	P	
REQ099	The solution SHOULD NOT provide a customised monitoring system as this will be performed by the future event management project	Y	
REQ100	The solution SHOULD provide security monitoring	Y	
REQ101	The solution MUST provide DNS and NTP service on the Active Directory Domain Controllers	Y	
REQ102	The solution MUST utilise the existing model for external DNS and NTP usage	Y	
REQ103	The solution MUST provide dual ExpressRoute paths one from each of the datacentres	P	
REQ104	The solution MUST provide ExpressRoute private peering	Y	
REQ105	The solution MUST provide ExpressROute redundancy for availability	Y	
REQ106	The solution MUST provide firewall architecture for public facing services in Azure	Y	
REQ107	The solution MUST NOT utilise the Cisco cloud web security proxy for egress HTTP/S traffic	Y	
REQ108	The solution MUST support a cloud datacentre as a trusted datacentre	Y	
REQ109	The solution MUST provide connectivity to the AV service layer (ESB)	Y	
REQ110	The solution MUST provide geo redundancy for the Azure edge DNS and NTP services i.e. hosted in Sydney as well	Y	3 NTP servers independent, anything dmz and non-windows will point to dmz ntp and windows will point to domain controllers
REQ111	The solution MUST provide integration with other AV applications	Y	
REQ112	The solution MUST provide network connectivity for IaaS to ESB, Oracle financials, VACIS, Adastra, ESTA, Optima Live and rostering	Y	
REQ113	The solution MUST provide a private network connection to each Azure DC (e.g. Telstra Cloud Gateway etc)	Y	
REQ114	The solution MUST provide geographically diverse dual path connectivity to the cloud provider	Y	

REQ115	The solution MUST provide highly available network architecture with no single point of failure i.e. dual paths	Y	
REQ116	The solution MUST provide network segregation and security control points across internal and external	Y	
REQ117	The solution MUST provide a routable subnet range that is within the AV allocation	P	
REQ118	The solution SHOULD provide connectivity to the edge DNS and NTP services to be hosted in Azure DMZ	Y	
REQ119	The solution SHOULD provide dual Telstra Cloud Gateway services where possible	Y	
REQ120	The solution SHOULD provide ExpressRoute termination on Azure within both Mel and Syd	Y	
REQ121	The solution SHOULD provide flexibility in the offering over cost concerns	Y	
REQ122	The solution SHOULD provide flexible bandwidth	Y	
REQ123	The solution SHOULD provide provide scalable bandwidth with minimum of 100MB for both ExpressRoutes	Y	
REQ124	The solution SHOULD utilise Vic health price list from Telstra	P	
REQ125	The solution SHOULD NOT provide internet access for the Azure virtual datacentre back via the on-premise internet connectivity	Y	
REQ126	The solution SHOULD provide secure internet connectivity via the Azure default internet gateway	Y	
REQ127	The solution SHOULD provide the ability to only allow third-parties to view non-prod but not production	Y	
REQ128	The solution SHOULD utilise the existing Impreva WAF or provide native equivalent or better functionality	Y	Impreva doesn't have load balancing capabilities
REQ129	The solution SHOULD NOT provide VPN support for third-parties	Y	
REQ130	The solution SHOULD provide a public ingress traffic load balancing capability	Y	
REQ131	The solution SHOULD provide an internal load balancing capability within the cloud environment	Y	
REQ132	The solution SHOULD provide an adequate level of network separation	Y	
REQ133	The solution SHOULD provide data transfers from production to non-production	Y	
REQ134	The solution SHOULD provide integration into selected existing onpremise services	Y	
REQ135	The solution SHOULD utilise SendGrid email raly service if needed	Y	
REQ136	The solution SHOULD provide a basic network zoning model	Y	
REQ137	The solution SHOULD NOT provide a dedicated firewall between Azure and the datacentre	Y	
REQ138	The solution SHOULD provide more segregation between production and non-production	Y	
REQ139	The solution SHOULD provide subnet level access controls for internal networks i.e. NSGs	N	Descoped as the AV network zoning model does not require this level of security
REQ140	The solution SHOULD provide subnet-based layer 4 access controls	Y	

REQ141	The solution MAY provide a third hybrid connectivity using VPNs for extra carrier redundancy	N	
REQ142	The solution MAY provide traffic flow optimisation	N	
REQ143	The solution MUST provide the SCCM agent on the base Windows server deployment to allow further configuration	Y	
REQ144	The solution MUST utilise WSUS for windows server updates	Y	
REQ145	The solution SHOULD utilise SCCM as a configuration management reporting tool	Y	
REQ146	The solution SHOULD provide an extensible and centralised reporting capability	Y	
REQ147	The solution SHOULD provide the ability to report on the network services	Y	
REQ148	The solution SHOULD provide the ability to report through Power BI	Y	
REQ149	The solution MAY provide technology for an analytics capability	N	Key project objective is providing HA and DR patterns for use so analytics is an optional rather than a mandatory requirement.
REQ150	The solution MUST utilise Windows Defender deployed by SCCM for virtual machines	Y	
REQ151	The solution MUST utilise the existing internal Certificate Authority housed in SDC for internal certificates	Y	
REQ152	The solution MUST provide a DMZ zone to host the extension of the existing external DNS and NTP edge services	Y	
REQ153	The solution MUST utilise the existing DMZ account management process which is local accounts on the VMs	Y	
REQ154	The solution MUST NOT provide an Azure Key Vault as it is out-of-scope - keys to be centralised into the enterprise PAM tool | dependant on the solution to be decided	Y	
REQ155	The solution SHOULD utilise the existing Tenable Vulnerability Scanning solution	Y	
REQ156	The solution SHOULD provide Azure Security Centre basic solution for reporting	Y	
REQ157	The solution SHOULD utilise the existing external certificate process via Thwart. Process is manual provisioning of certificates	Y	
REQ158	The solution MUST provide a non-prod and production environment structure aligned to AV's default delivery pipeline	Y	
REQ159	The solution MUST provide environmental categorisation for Production and DR as "Production"	Y	
REQ160	The solution MUST provide dual Azure datacentres for resilience/DR	Y	
REQ161	The solution SHOULD provide a recommended maintenance window	Y	
REQ162	The solution SHOULD provide dual Azure datacentres preferably both in Victoria for data cost management	P	
REQ163	The solution SHOULD support migration of existing Azure assets and decommissioning of current subscriptions	Y	

Appendix B - Contents Tables

Table of Key Design Decisions

Key Design Decision 3-1: Environment Design 9
Key Design Decision 3-2: Environment OLAs 10
Key Design Decision 3-3: Environment Maintenance Windows 11
Key Design Decision 4-1: Subscription Partitioning 13
Key Design Decision 4-2: Azure Regions 14
Key Design Decision 4-3: Resource Groups 15
Key Design Decision 4-4: Resource Tagging 15
Key Design Decision 4-5: Cost Management 15
Key Design Decision 4-6 - Reserved Instances 16
Key Design Decision 5-1: Wide Area Network 18
Key Design Decision 5-2: Telstra Cloud Gateway 19
Key Design Decision 5-3: ExpressRoute 22
Key Design Decision 5-4: IaaS Internet Traffic 22
Key Design Decision 5-5: PaaS Internet Traffic 22
Key Design Decision 5-6: Imperva Incapsula Global Server Load Balancing 22
Key Design Decision 5-7: F5 BIG-IP DNS Service 23
Key Design Decision 5-8: Imperva Incapsula WAF 25
Key Design Decision 5-9: Cisco Adaptive Security Appliance virtual (ASAv) Firewall 26
Key Design Decision 5-10: Virtual Networks 28
Key Design Decision 5-11: Virtual Network Peering 28
Key Design Decision 5-12: Virtual Network Gateway 29
Key Design Decision 5-13: Subnets 30
Key Design Decision 5-14: Subnet Sizing 30
Key Design Decision 5-15: Network Security Groups 31
Key Design Decision 5-16: Virtual Network Service Endpoints 31
Key Design Decision 5-17: Internal Load Balancer 31
Key Design Decision 5-18: Public Load Balancer 32
Key Design Decision 5-19: DNS Services 32
Key Design Decision 5-20: User-Defined Routes 32
Key Design Decision 6-1: Storage Accounts 34
Key Design Decision 6-2: Storage Account Access Control 35
Key Design Decision 6-3: Storage Account Encryption 35
Key Design Decision 6-4: Managed Disks 36
Key Design Decision 6-5 - Monitoring and Alerts 36
Key Design Decision 6-6: Commvault 37
Key Design Decision 6-7: Azure Site Recovery Service 39
Key Design Decision 6-8: Site Recovery Deployment Planner 40
Key Design Decision 6-9: ASR Resource Group 40
Key Design Decision 6-10: Storage Account 40
Key Design Decision 6-11: Recovery Service Vault for ASR 40
Key Design Decision 6-12: Azure Recovery Service Network Configuration 40
Key Design Decision 6-13: On-premise Configuration Server 41
Key Design Decision 6-14: Azure SQL Database 43
Key Design Decision 6-15: Azure SQL Replication 43

Key Design Decision 6-16: Azure Automation 44
Key Design Decision 7-1: Azure Security Center 46
Key Design Decision 7-2: Role Based Access Control (RBAC) 48
Key Design Decision 8-1: Active Directory 50
Key Design Decision 8-2: AD Sites & Services 50
Key Design Decision 8-3: Azure Active Directory 51
Key Design Decision 8-4: AADC 51
Key Design Decision 8-5: ADFS 51
Key Design Decision 8-6: Azure Enterprise Applications 52
Key Design Decision 8-7: MFA 52
Key Design Decision 9-1: Virtual Machine Specifications 53
Key Design Decision 9-2: Virtual Machine Operating Systems 53
Key Design Decision 9-3: Windows Updates 54
Key Design Decision 9-4: Azure Virtual Machine Extensions 54
Key Design Decision 9-5 - Azure Virtual Machine Configuration for Linux 54
Key Design Decision 9-6: Virtual Machine Antimalware Protection 55
Key Design Decision 9-7: Azure Application Service Environments 56
Key Design Decision 9-8: Azure Functions 57
Key Design Decision 9-9: API Management 57
Key Design Decision 10-1: Deployment Tools 59
Key Design Decision 10-2: Infrastructure Coding Language 60
Key Design Decision 10-3: Code Repository and Source Control (Git) 60
Key Design Decision 10-4: Pull Request Approvals 61
Key Design Decision 10-5: Continuous Integration 62
Key Design Decision 10-6: CD Environments 62
Key Design Decision 10-7: Release Triggers and Approvals 63
Key Design Decision 10-8: VSTS Identity and Access Management 63
Key Design Decision 10-9: Azure Service Integration 64

Table of Recommendations

Recommendation 4-1: Cost Management 15
Recommendation 6-1: Azure Backup 37
Recommendation 9-1: Windows Updates 54
Recommendation 9-2 - Virtual Machine Antimalware Protection 55
Recommendation 9-3: Azure App Service 56
Recommendation 9-4: API Management 57
Recommendation 9-5: Azure Service Fabric 58
Recommendation 10-1: Code Commits 61

Table of Figures

Figure 1 - System Context Diagram 7
Figure 2 - Environment Classification Mapping 9
Figure 3 - EA Enrolment Infographic 12
Figure 4 - Ambulance Victoria Azure EA Structure 13
Figure 5 - Azure Reserved Instances Benefits Diagram 16

Figure 6 – Proposed Wide Area Network Overview 17
Figure 7 - Telstra Cloud Gateway connectivity options 18
Figure 8 - Telstra Cloud Gateway georedundancy 19
Figure 9 - ExpressRoute Overview 20
Figure 10 - Network Firewall Reference Architecture 24
Figure 11 - ASAv Reference Architecture 26
Figure 12 - Azure Internal Network Diagram 27
Figure 13 - ASR Architectural Diagram 38
Figure 14 - ASR Architectural Components 39
Figure 15 - Configuration Server Sizing Recommendations 41
Figure 16 - Azure SQL Replication Model 43
Figure 17 - Active Directory Sites and Services 50
Figure 18 – Virtual Machine Specification Mapping 53
Figure 19 - Azure Service Fabric Infographic 58
Figure 20 - Release Process Diagram 60
Figure 21 - Traceability Table by the Criticality of the Requirement 65